Alibaba Cloud
Resource Access Management

User Guide

Issue: 20181105

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.
5. By law, all the content of the Alibaba Cloud website, including but not limited to works, products, images, archives, information, materials, website architecture, website graphic layout, and webpage design, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade

secrets. No part of the Alibaba Cloud website, product programs, or content shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates).

6. Please contact Alibaba Cloud directly if you discover any errors in this document.

Generic conventions

Table -1: Style conventions

Style	Description	Example
	This warning information indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	 **Danger:** Resetting will result in the loss of user configuration data.
	This warning information indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	 **Warning:** Restarting will cause business interruption. About 10 minutes are required to restore business.
	This indicates warning information, supplementary instructions, and other content that the user must understand.	 **Note:** Take the necessary precautions to save exported data containing sensitive information.
	This indicates supplemental instructions, best practices, tips, and other content that is good to know for the user.	 **Note:** You can use **Ctrl** + **A** to select all files.
>	Multi-level menu cascade.	**Settings** > **Network** > **Set network type**
Bold	It is used for buttons, menus, page names, and other UI elements.	Click **OK**.
`Courier font`	It is used for commands.	Run the `cd /d C:/windows` command to enter the Windows system folder.
`Italics`	It is used for parameters and variables.	`bae log list --instanceid` *`Instance_ID`*
[] or [a|b]	It indicates that it is a optional value, and only one item can be selected.	`ipconfig` *`[-all|-t]`*
{} or {a|b}	It indicates that it is a required value, and only one item can be selected.	`swich` *`{stand | slave}`*

Contents

1 Overview

This document mainly describes the core functions of RAM and their typical application scenarios.

Core functions

Identity management

- *User*
- *Group*
- *Role*

Authorization management

- *Authorization Policy Management*
- *Authorization*
- *Policy syntax structure*

Typical application scenarios

- *RAM user management and authorization for enterprises*
- *Temporary authorization management of mobile apps*
- *Cross-account resource access and authorization*

2 Identities

2.1 User

RAM user is an identity used in RAM to relate with a true identity, such as a user or an applicatio n. To allow a new user or a new application to access your cloud resources, you can create and grant permissions to a RAM user. The general procedure is as follows:

1. Use the primary account (or a RAM user with RAM operation permissions) to log on to the *RAM console*.
2. Create a RAM user and add the user to one or more groups.
3. Attach one or more authorization policies to the user (or the group to which the user belongs).
4. Create a credential for the user. If the user performs operations through the console, set a logon password for the user. If the user performs operations by calling APIs, create an API AccessKey for the user.
5. If the user needs to use special permissions (for example, to stop ECS instances), you can set MFA for the user and require that the user use an MFA password to log on to the Alibaba Cloud console.
6. Provide the user with the logon URL, username, and password.

RAM settings

- *Set the enterprise alias*
- *Configure the password policy*
- *Configure the security policy*

Set the enterprise alias

1. In the **RAM console**, choose **Settings** > **Enterprise Alias Settings** > **Edit Enterprise Alias**.
2. Enter an **enterprise alias** and click **OK**.

Configure the password policy

To configure the password policy, follow these steps:

1. In the **RAM console**, choose **Settings** > **Password Strength Settings**.
2. Follow the page prompts to configure rules such as password length, character format, expiration date, and retry constraint policy, and then click **Save Changes** to make the rules take effect.

Note:
Once the password policy takes effect, all RAM users created hereafter must comply with the password strength settings.

Configure the security policy

1. In the **RAM console**, choose **Settings** > **User Security Settings**.
2. On the **User Security Settings** page, configure your security policy.
3. Click **Save Changes**.

Create a RAM user

To create a RAM user, follow these steps:

1. In the **RAM console**, choose **Users** > **New User**.
2. Enter user information in the displayed dialog box and click **OK**.

After creating a RAM user, you can do the following as needed:

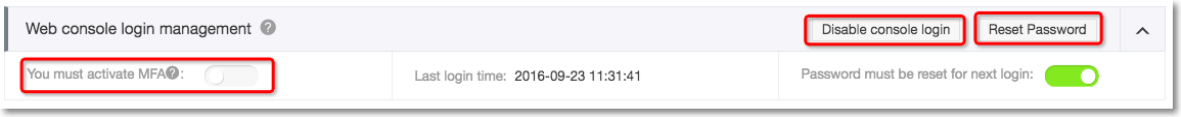
- *Set a logon password*
- *Create an AccessKey*
- *Enable virtual MFA devices*

Set a logon password

To allow a RAM user to access the **RAM console**, you can set a logon password for the user. The procedure is as follows:

1. Click **Users** in the left navigation bar of the **RAM console**. Select the target user (which can be queried by user name). Click the user name or **Manage**.
2. On the **User Details** page, click **Enable Console Logon**.

Figure 2-1: Set a logon password



3. In the displayed dialog box, set an initial password for the user. You can specify a rule that the user must change the password upon logon.

Create an AccessKey

To create an AccessKey for the user who needs to call the API, follow these steps:

1. Click **Users** in the left navigation bar of the RAM console. Select the target user (which can be queried by user name). Click the user name or **Manage**.
2. On the **User Details** page, click **Create Access Key**.
3. In the displayed dialog box, view the new AccessKey information and click **Save Access Key Information**.

Note:

- New AccessKeys are displayed only during creation. For security reasons, RAM does not provide an AccessKey query interface. Therefore, please keep the AccessKey safe.
- If your AccessKey is disclosed or lost, you must create a new one.

Enable virtual MFA devices

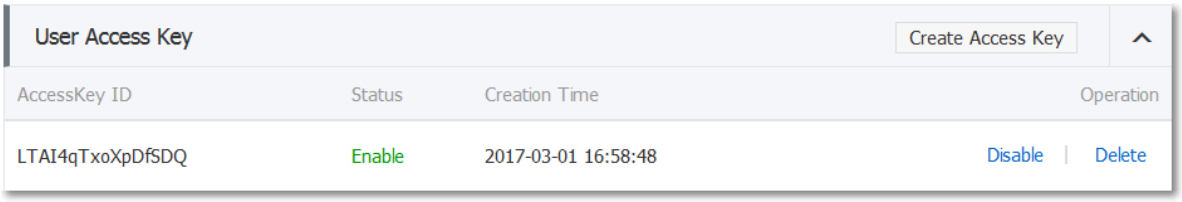
Multi-Factor Authentication (MFA) is a simple but effective best practice that can provide additional security protection. After MFA is enabled, when a user logs on to Alibaba Cloud, the system requires the user to enter the user name and password (first security factor), and then enter a variable verification code (second security factor) provided by the user's VMFA (virtual MFA) device. All these factors work together to offer higher security protection for your account.

The virtual MFA (VMFA) device is an application that generates a 6-digit verification code. It complies with the time-based one-time password algorithm (TOTP) standard (*RFC 6238*). This application can run on mobile hardware devices including smartphones, making it easily accessible.

To enable virtual MFA devices for a RAM user, follow these steps:

1. Click **Users** in the left navigation bar of the **RAM console**. Select the target user (which can be queried by user name). Click the user name or **Manage**.
2. On the **User Details** page, click **Enable VMFA Device** to start the *Set up MFA (optional)* procedure.

Figure 2-2: Set MFA



RAM user logon

Logon entry

RAM user and cloud account logon entries are different. RAM users cannot log on through the cloud account logon page.

The RAM user's logon link is *https://signin.aliyun.com/login.htm* (The logon link can be queried through the **Dashboard** page of the *RAM console*.)

Login information

RAM User logon requires an enterprise alias, a sub-user name, and a password.

The enterprise alias is the one you have set in the initial RAM setup. If you have not set an enterprise alias, the default enterprise alias is your cloud account ID (which can be queried through**Account Management** > **Security Settings**.

Note:
By default, RAM users do not have any access permissions. A RAM user without permissions can log on to the console, but cannot perform any operations. For details about how to attach a policy to a RAM user, see *Authorization*.

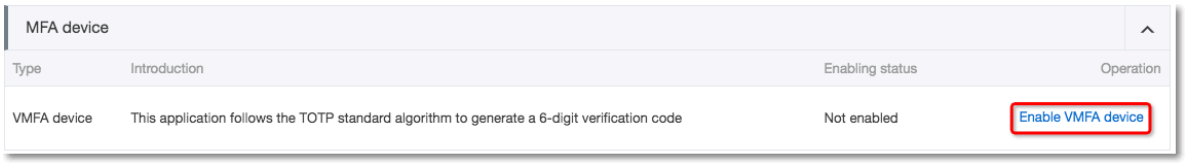
Delete a RAM user

Note:
Think it over before deleting a RAM user. If a user is running a certain service, deleting this user may cause a service failure.

To delete a RAM user, follow these steps:

1. Click **Users** in the left navigation bar of the RAM console to open the **User Management** page.
2. Find the RAM user you want to delete and click **Delete** in the **Actions** area.

Figure 2-3: Delete a RAM user



3. In the displayed **Delete User** dialog box, select the **Unlink Dependent Objects** check box, and then click **OK**.

Figure 2-4: Confirm the deletion

2.2 Group

If you have created multiple RAM users with your Alibaba Cloud account, we recommend that you manage those users by group to simplify the management process. Classify RAM users with the same responsibility by group and *Authorization* upon authorization. The benefits of doing so are as follows:

- When the responsibility of a specific user changes, you can simply move them to an appropriat e group, which does not have an impact on other users.
- When the permissions of a group change, you only need to modify the authorization policy for the group, which can be applied to all users.

This document describes the following tasks:

- *Create a group*
- *Manage group members*
- *Rename a group*
- *Delete a group*
- *Grant permissions to a group*

Create a group

The operation procedure is as follows:

1. On the homepage of the *RAM console*, click**Groups** > **Create Group**.
2. Enter a Group Name and click **OK**.

Manage group members

The operation procedure is as follows:

1. On the homepage of the **RAM console**, click Groups.
2. On the **Group Management** page, click **Management** in the **Actions** area to manage group members.

- Add group members:
  1. Locate the group you want to manage in the group list (you can use the group name for a fuzzy query), and click **Edit Group Member** in the Actions area.
  2. Select the target user from the left box (you can query using the keyword), and click the right arrow to add the user to the right box; select the user in the right box, and click the left arrow to undo the selection.
  3. Click **OK** when the selection is complete.
- Delete group members:
  1. Locate the group you want to manage in the group list (you can use the group name for a fuzzy query), and click the group name.
  2. In the **Group Member Management** area, click **Remove from Group** to delete the target user from the group.

Rename a group

The operation procedure is as follows:

1. On the homepage of the **RAM console**, click **Groups**.
2. Locate the group you want to rename in the group list (you can use the group name for a fuzzy query), and click the group name or **Management** to enter the **Group Details** page.
3. Click **Edit Basic Info**.
4. Enter **Group Name** and click **OK**.

Delete a group

The operation procedure is as follows:

1. On the homepage of the **RAM console**, click **Groups**.
2. Locate the group you want to delete in the group list (you can use the group name for a fuzzy query), and click **Delete** in the Actions area.

   Note:
   If the group contains group members or a bound authorization policy, you can delete a group only after selecting **Unlink Dependent Objects**.

Grant permissions to a group

For details about how to grant permissions to a group, see *Authorization*.

2.3 Role

Like a RAM-User, a RAM-Role is also a type of RAM identity. Compared with a RAM-User, a RAM -Role is a virtual user, that is, a RAM-Role has no identity credentials and has to be assumed by a trusted Alibaba Cloud account.

With this document, you can gain a better understanding of the RAM-Role, and know how to create and use a RAM-Role.

Note:
Unless otherwise stated, the **role** in this document represents a **RAM-Role**.

Understanding RAM-Role

A RAM-Role is a virtual user (or shadow account). It is a type of RAM identity.

- A RAM-Role differs from a Textbook-Role. A Textbook-Role (or a role as traditionally defined) indicates a set of permissions. It is similar to a policy in RAM. If a role is granted to a user, this means that the corresponding permissions are granted to the user. Then, this user can access the authorized resources.
- As a type of virtual user, a RAM-Role has a fixed identity and can be granted policies. However, it does not have a fixed security credential (such as a logon password or an AccessKey).

  Virtual users vs. Real users: The difference between a virtual user and a real user is that a real user identity can be directly authenticated.

  - A real user has a logon password or an AccessKey. For example, Alibaba Cloud accounts, RAM-User accounts, and cloud service accounts are real users.

  - However, a virtual user, such as a RAM-Role, does not have a fixed security credential.
- RAM-Roles must be assumed by an authorized real user. After assuming a role, the real user receives a temporary security token for this RAM-Role. Then, the user can use this temporary security token to access the resources authorized for the role.

Notes

A RAM-Role must be **associated** with a real user identity so that it becomes available.

- If a real user wants to use a RAM-Role that has been granted to the user, the real user must first log on using his identity and then perform the **SwitchRole** operation to switch from **a real identity** to **a role identity**.

   **Note:**
  The user can then perform all operations authorized for this role identity, but the access permissions of the user's real identity will not be available.
- To switch from **the role identity** back to **the real identity**, the user must perform the **Switch Back to Logon Identity** operation.

   **Note:**
  Then, the user can have the access permissions granted to his real identity, but not those of the role.

Concepts

The following table lists several basic concepts related to RAM-Roles:

Concept	Meaning
RoleARN	A RoleARN is the global resource description of a role. It is used to specify a role. • RoleARNs follow Alibaba Cloud ARN naming rules. For example, the RoleARN for the devops role under an Alibaba Cloud account is: `acs:ram::1234567890123456:role/samplerole`. • After a RAM-Role is created, the role's ARN is displayed on the **Role Details** page.
Trusted Actors	A role's trusted actors are the real user identities (the current Alibaba Cloud account

Concept	Meaning
	or another Alibaba Cloud account) that can assume this role. • When creating a role, you must specify the trusted actors. A role can only be assumed by trusted actors. • A trusted actor can be a trusted cloud account or a trusted service.
Policy	A role can be attached with a set of permission s, that is, a policy. Roles not attached with policies can exist, but cannot be used.
Assume Role	By performing the assume role operation, a real user can obtain a security token for a role . By calling the AssumeRole API, a real user obtains the role’s security token and can use this token to access cloud service APIs.
Switch Role	By performing the switch role operation on the console, a real user can switch from the current logon identity to a role identity. • After a real user logs on to the console, the user can switch to a role for which he is a trusted actor. Then, the user can use the role identity to perform operations on cloud resources. After switching to a role identity, the user's real identity access permissions are no longer available. • When the user no longer needs to use a role, he can switch from the role back to the original logon identity.
Role Token	A role token is a temporary AccessKey for the role identity. Role identities do not have fixed AccessKeys. Therefore, when a real user wants to use a role, he or she must assume the role to obtain the corresponding role token . Then, the user can use this role token to call Alibaba Cloud service APIs.

Application scenarios

RAM-Role is mainly used to entrust other cloud accounts and their RAM users as well as cloud services to operate on your cloud resources.

Cross-account resource operation and authorization management

Scenario description: Enterprise A and Enterprise B represent two different companies. Enterprise A has purchased a variety of cloud resources (for example, ECS instances, RDS instances, SLB instances, OSS buckets, etc.) to conduct business.

Requirement	Solution
Enterprise A wants to focus on its business systems, so it authorizes Enterprise B to perform the tasks of maintaining, monitoring, and managing cloud resources.	Alibaba Cloud account A creates a role in RAM and grants this role the necessary permissions. Then, it allows Alibaba Cloud account B to use this role.
Enterprise B assigns the entrusted maintenance tasks to its employees. B needs to precisely control the permissions assigned to its employees who operate on A's resources.	If account B has employees (RAM-Users) who need to use this role, it can independently control their permissions. When performing O&M operations on behalf of A, account B’s RAM-users can use the role identity to perform operations on A’s resources.
If A and B terminate this O&M entrustment contract, enterprise A is able to revoke the permissions of the enterprise B as needed.	If accounts A and B terminate their contract, A just needs to revoke B’s permission to use this role. Once account B’s permission to use this role is revoked, all RAM-Users of account B will automatically lose their permission to use this role.

Temporary authorization access

Scenario description: Assume that Enterprise A has developed a mobile app and has bought OSS. The mobile app must upload and download data to and from OSS, but A does not want to allow all apps to use the AppServer to transmit data. Because the mobile app runs on user devices, these devices are out of A’s control. For security reasons, A cannot save the AccessKey in the app.

Requirement	Solution
Enterprise A does not want all apps to transmit data through the AppServer. Instead, A	• Alibaba Cloud account A creates a role in RAM and grants this role the necessary permissions. Then, it allows the AppServer (

Requirement	Solution
wants to allow the apps to directly upload and download data to and from OSS.	giving it a RAM user identity) to use this role . • When the app needs to directly connect to OSS to upload and download data, the AppServer can use this role to obtain the role's temporary security token and send it to the app. The app can then use the temporary security token to directly access OSS APIs.
Enterprise A wants to minimize its security risks by, for example, giving each app an access token with only the minimum permissions it needs when directly connected to OSS and restricting the access duration to a short period of time (such as 30 minutes).	• If more precise control over the permissions of each app is required, when using the role, the AppServer can further restrict the resource operation permissions of the temporary security token. • For example, when assuming the role, the AppServer can restrict that different app users can perform operations only on a few subdirectories.

Entrust cloud services to operate on your cloud resources

Scenario description: Enterprise A has purchased the ECS server and has deployed an application in it. The application needs to access A's OSS bucket. In most cases:

- Cloud account A saves its AccessKey (AK) in the application's configuration file, and modifies the file when periodically replacing the AK.
- In the case of multi-region consistency deployment, the AK is spread out with the images and the instances created with the images. If this happens, when A needs to replace the AK, it is necessary to update and redeploy the instances and images one by one.

Requirement	Solution
• For security reasons, A does not want its application to get full access to its API operations through the AK. Instead, A expects the application to access the APIs of other products with the STS. • For operational considerations, A does not want to update the AK on the application	With the RAM service role: • Cloud account A creates an ECS service role (assumed by ECS instances only) in RAM, grants appropriate permissions to the role (such as read-only access to OSS), and associates the service role with its ECS instance.

Requirement	Solution
, and does not want to maintain the AK in multiple regions.	• After the ECS instance is connected, the STS of the service role is obtained by accessing the metadata of the ECS instance . The application in the ECS server uses the STS to access the OSS.

Note:

The RAM service role can be used for authorization operations in cross-product scenarios, such as authorizing the EMR to operate on the customer's ECS, the FC to operate on the customer's OSS, and the MTS to operate on the customer's OSS data. For all service role types and scenarios provided by RAM, see *Create a service role*.

About the PassRole permission for RAM

In order to limit what a service can do on your behalf, you need to configure a RAM role for the service. The service will perform the relevant operations as the corresponding RAM role and is limited by the administrator's authorization for the RAM role. For example, you can configure a RAM role for an ECS instance, and the application in the ECS instance can obtain the STS of the corresponding RAM role to access Alibaba Cloud APIs.

When a RAM user configures a RAM role for a cloud service, the RAM user must have the PassRole permission for the RAM role. Once the cloud service receives a request for RAM role configuration, a mandatory check will be performed to see if the RAM user has the ram:PassRole permission for the specified RoleArn. This ensures that only authorized users can configure RAM roles for cloud services, without causing abuse of RAM role permissions.

RAM role types

RAM supports two types of roles:

- **User role**: role that can be assumed by RAM users. RAM users permitted to assume roles can belong to your Alibaba Cloud account or another Alibaba Cloud account. User roles are used to solve problems such as cross-account access and temporary authorization access.
- **Service role**: role that can be assumed by cloud services. Service roles are used to authorize cloud services to perform operations on resources on your behalf.

Create a RAM role

To create a RAM role in the RAM console, perform the following steps:

1. Select the role type.
2. Select the trusted actor.
3. Enter the role name.
4. Bind an authorization policy to the role.

Create a user role

The procedure is as follows:

1. Log on to the *RAM console*.
2. In the left navigation pane, click **Roles**.
3. On the **Role Management** page, click **Create Role**.
4. On the **Select Role Type** page, click **User Role**.
5. On the **Enter Type Information** page, do one of the following and click **Next**.
   - If the role is to be used by the RAM-Users under your own account (such as authorizing a mobile app client to directly perform operations on OSS resources), select your **Current Alibaba Cloud Account** as the trusted Alibaba Cloud account.
   - If the role is to be used by the RAM-Users under another Alibaba Cloud account (such as for cross-account resource authorization access), select **Other Alibaba Cloud Account** and enter its ID in the **Trusted Alibaba Cloud Account ID** field.
6. On the **Configure Basic Role Information** page, enter a **Role Name** (the description is optional) and click **Create**.
7. After you have successfully created a role, you can click **Authorize** to grant permissions to the role or click **Close**. For details, see *Authorization*.

Now you have created the user role.

Go back to the *Role Management* page and you can find the newly created role in the role list. Click the **Role Name** or the corresponding **Manage** in the **Actions** column to enter the **Role Details** page, where you can find the role's ARN and can **Edit Basic Information**.

Create a service role

The procedure is as follows:

1. In the RAM console, click **Roles** in the left navigation pane.
2. On the **Role Management** page, click **Create Role**.
3. On the **Select Role Type** page, click **Service Role**. Available service roles include the following:

- MTS Media Transcoding. When setting the OSS Bucket to the data source for the MTS service, create a role with MTS as the trusted service and use the MTS service to assume this service role to access data in the OSS.
- OAS Archive Storage. When setting the OSS Bucket to the data source for the archive storage service, create a role with archive storage as the trusted service and use the archive storage service to assume this service role to access data in the OSS.
- LOG Log Service. When importing logs collected by the log service into the OSS, create a role with log service as the trusted service and use the log service to assume this service role to write data into the OSS.
- ApiGateway Service. When setting the function service to a backend service for an API gateway, create a role with API gateway as the trusted service and use the API gateway service to assume this service role to call the function service.
- ECS Elastic Compute Service, which is used to authorize the ECS service to access your cloud resources in other cloud services.

4. On the **Enter Type Information** page, select a service as the trusted service.
5. On the **Configure Basic Role Information** page, enter a **Role Name** (the description is optional) and click **Create**.
6. After you have successfully created a role, you can click **Authorize** to grant permissions to the role or click **Close**. For details, see *Authorization*.

Now you have created the service role.

Go back to the *Role Management* page and you can find the newly created role in the role list. Click the **Role Name** or the corresponding **Manage** in the **Actions** column to enter the **Role Details** page, where you can find the role's ARN and can **Edit Basic Information**.

Use a RAM role

A RAM role can only be assumed by RAM users in the trusted Alibaba Cloud account. For security reasons, the trusted Alibaba Cloud accounts are not allowed to perform AssumeRole.

Therefore, you must use a trusted account to create a RAM-User account, and grant the AssumeRole permission to the RAM-User account. Then, you can assume the role by using this RAM-User identity.

The procedure is as follows:

1. Create a RAM-User and create an AccessKey or set a logon password for this user.

2. Grant the following system authorization policy to this RAM-User: `AliyunSTSAssumeRoleAccess`

Use a RAM role to perform console operations

If a RAM-User needs to use the role identity to perform console operations, the RAM-User must first log on to the console with the logon identity, and then use the **SwitchRole** method. After that, the user can use the role identity to perform console operations.

For example, the RAM-User Alice under company2 (enterprise alias) logs on to the console, the user can move the mouse pointer to the account name on the upper-right corner and click **Switch Role**.

- Alice needs to select the corresponding company alias and role name. For example, we assume that the user has been granted permission to assume the **ecs-admin** role of company1 (enterprise alias).
- After switching to the role, Alice can use the role identity to access the console.

Use a RAM role to access APIs

After a RAM-User is granted the AssumeRole permission, the user can use the AccessKey to call the STS AssumeRole API to obtain a temporary security token for this role. For details about how to call the AssumeRole API, see API documentation.

3 Authorization

3.1 Permissions and authorization policies

Alibaba Cloud uses permissions to describe the ability of internal identities (such as users, user groups, roles) to access specific resources. Permission refers to **allowing** or **denying** to perform certain operations on certain resources under certain conditions. An authorization policy is a collection of access permissions.

This document describes the relationships between Alibaba Cloud permissions and authorization policies, to help you understand and use them correctly.

Permissions

- The primary account (resource owner) controls all permissions.
  - Each resource has one and only one owner (resource owner). The owner must be an Alibaba Cloud account. This account pays for and has full control over the resource.
  - The resource owner is not necessarily the resource creator. For example, a RAM user is granted permission to create a resource. The resource belongs to the primary account. This user is the resource creator but not the resource owner.
- A RAM user (operator) does not have any permissions by default.
  - A RAM user represents an operator and must be explicitly authorized to perform any operation.
  - A new RAM user has no operation permission by default, and cannot operate on resources in the console or by using APIs until the authorization is granted.
- Resource creators (RAM users) are not automatically granted permissions for the resources they have created.
  - If a RAM user is granted permission to create a resource, the user will be allowed to create the resource.
  - However, the RAM user does not automatically have any permissions for the created resource, unless the resource owner has an explicit authorization on him.

Authorization policies

An authorization policy is a set of permissions described by *Policy syntax structure* that accurately describes the set of authorized resources, the set of operations, and the authorization conditions.

The **deny first** principle is followed when there are both allow and deny authorization statements in an authorization policy.

In RAM, an authorization policy is a type of resource entity. Users can create, update, delete, and view authorization policies. RAM supports the following two authorization policies:

- **System authorization policies** are a group of general permissions created and managed by Alibaba Cloud, such as read-only permission for ECS or full permissions for ECS. You can use these policies, but cannot modify them.
- **Custom authorization policies** are a group of permissions created and managed by users. They can be used to expand and supplement system authorization policies.

System authorization policies describe coarse-grained permissions. If you require finer-grained authorization policies, such as policies that precisely control permissions for a certain ECS instance or that have additional authorization conditions, you must create custom authorization policies.

Authorization to a RAM user

Authorization to a RAM user refers to binding one or more authorization policies to a user, user group, or role.

- The bound authorization policy can be either a system authorization policy or a custom authorization policy.
- If a bound authorization policy is updated, the updated policy automatically takes effect, and you do not have to rebind it.

3.2 Authorization Policy Management

Authorization Policy is a collection of permissions defined in ALI cloud.*Elements* by a more common deployment. By adding authorization policies for a user or a group, the user or all users in the group can acquire the access permission specified in the authorization policy.

Ram supports two types of authorization policies: System Authorization policies and *Custom Authorization Policy*. This paper introduces the management method of Authorization Policy, including: to view the System Authorization Policy, and, modify, and delete, customize authorization policies.

System authorization policy

System authorization policies are a set of generic authorization policies provided by AliCloud , specific to the read-only permission or all permissions for different products. For this set of licensing strategies offered by Ali cloud,

- Users can only be used for authorization, not editing and modification.
- Ali cloud updates or modifies automatically.

Viewing System Authorization policies

If you want to view all System Authorization policies supported by Ali cloud, log in to The **RAM console**, and enter the Authorization Policy Management page, under the System Authorization Policy subpage, view or search through the list of System Authorization policies.

Custom Authorization Policy

If the coarse-grained system authorization policies cannot meet your needs, you can create custom authorization policies. For example, you want to control a specific ECs The operation permission for the instance, or the resource operation request that you request from the visitor must come from the specified IP address, you must use a custom authorization policy to meet this fine-grained requirement.

Application scenarios

If you have more fine-grained authorization requirements, such as authorized user Bob can only pair OSSS: /sample_bucket/Bob/ All objects under perform read-only operations, and limiting IP sources must be part of your corporate network (through a search engine query, "My IP", you can learn about your corporate network IP Address), then you can access control by creating a custom Authorization Policy.

Create a custom Authorization Policy

When creating a custom Authorization Policy, you need to understand the basic structure and syntax of the authorization policy language, please refer to for a detailed description of the relevant content.

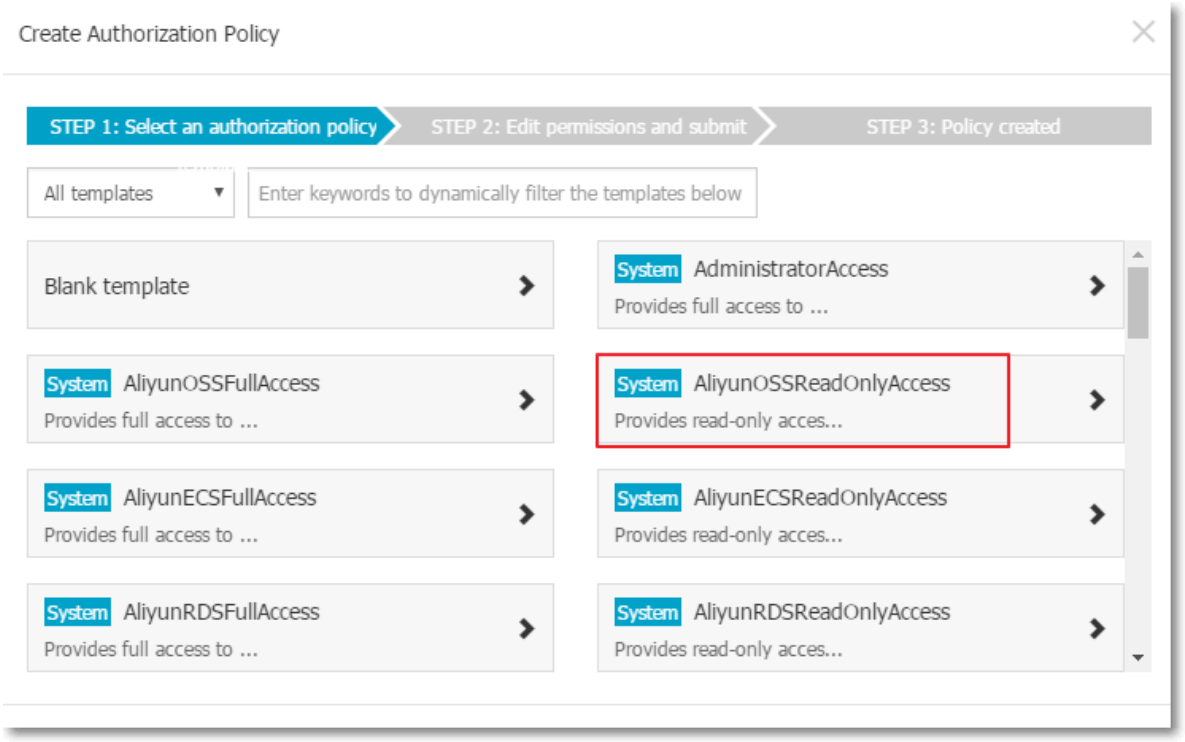
Operation Steps

After learning about the authorization policy language, you can easily create custom authorization policies to meet the above needs on the **RAM console**.

1. Log on to the *RAM console*.

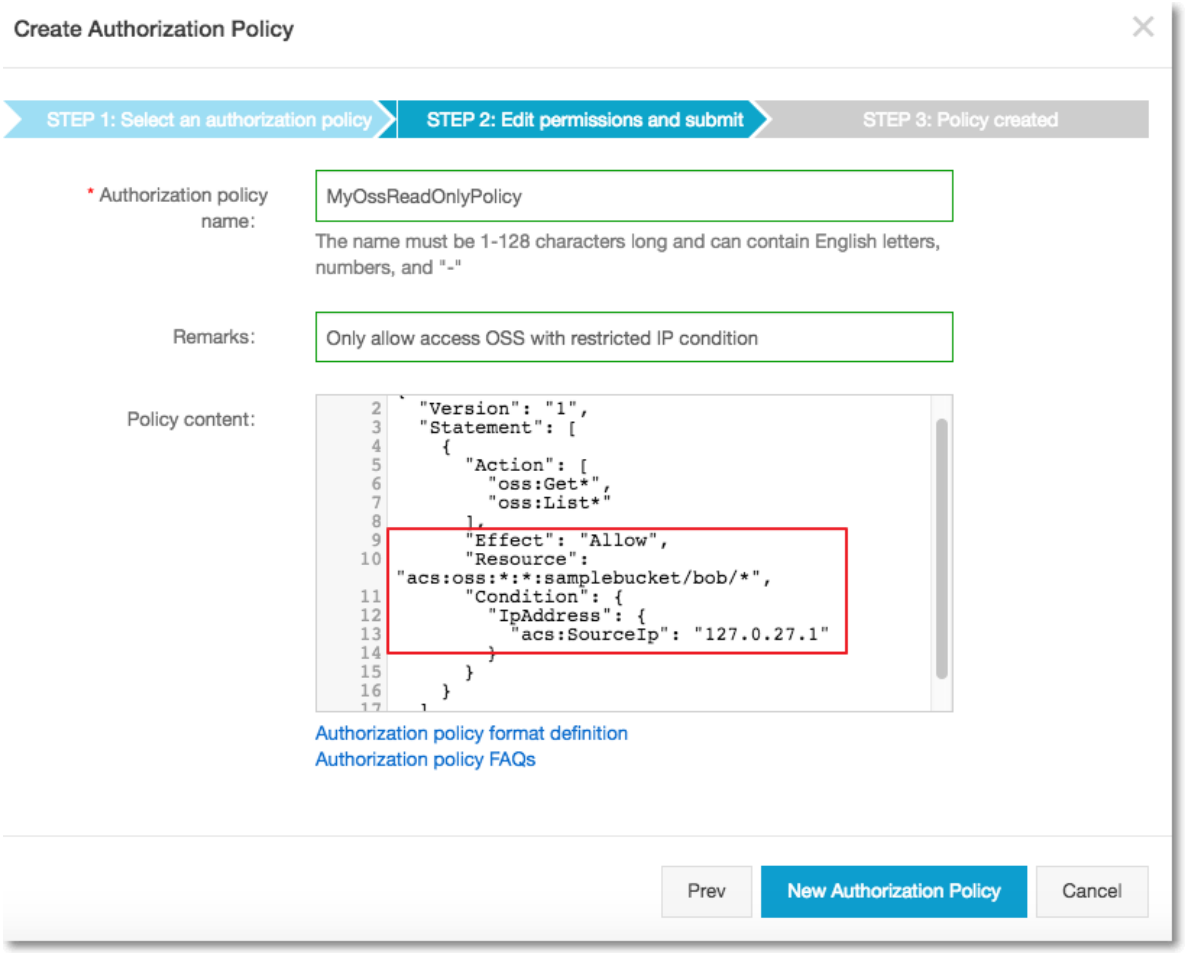
2. Click**Policy Management** > **Custom Authorization Policy**.
3. Click new authorization policy to open the new authorization policy burst window, as shown in the following figure:

Figure 3-1: New Authorization Policy



4. Select a template (for example, AliyunOSSReadOnlyAccess). Then, you can edit the policy based on the template, as shown in the figure below:

Figure 3-2: Edit Authorization Policy



The name, remarks, and content of the custom authorization policy have been modified. In the above figure, the selected part is the added fine-grained authorization content.

The template sample is as follows:

```
{
    Version: "1 ",
    "Statement ":[
      {
        "Action ":[
          "Oss: Get *",
          "Oss: list *"
        ],
        "Effect": "allow ",
        "Resource": "ACS: OSS: *: samplebucket/Bob /*",
        "Condition ":{
          "IPaddress ":{
            "ACS: sourceip": "maid"
          }
        }
      }
    ]
```

```
}
```

5. Click new authorization policy to complete the new custom Authorization Policy.

Subsequent operation

If you attach this custom Authorization Policy to user Bob, then Bob vs. OSS: // samplebucket/ Bob/ The object under has read-only operation permissions, and the restriction is that it must be accessed from your corporate network (assumed to be 121.0.257.1.

Please refer to the specific operation.

Modify a custom Authorization Policy

When a user's permissions change (new permissions are added or existing permissions are revoked), you must modify the user's authorization policy. When you modify an authorization policy, you may encounter the following problems:

- Hopefully, after some time, the old authorization policy will continue to be used.
- After the modification is complete, you find that the authorization policy has been modified incorrectly and needs to be rolled back.

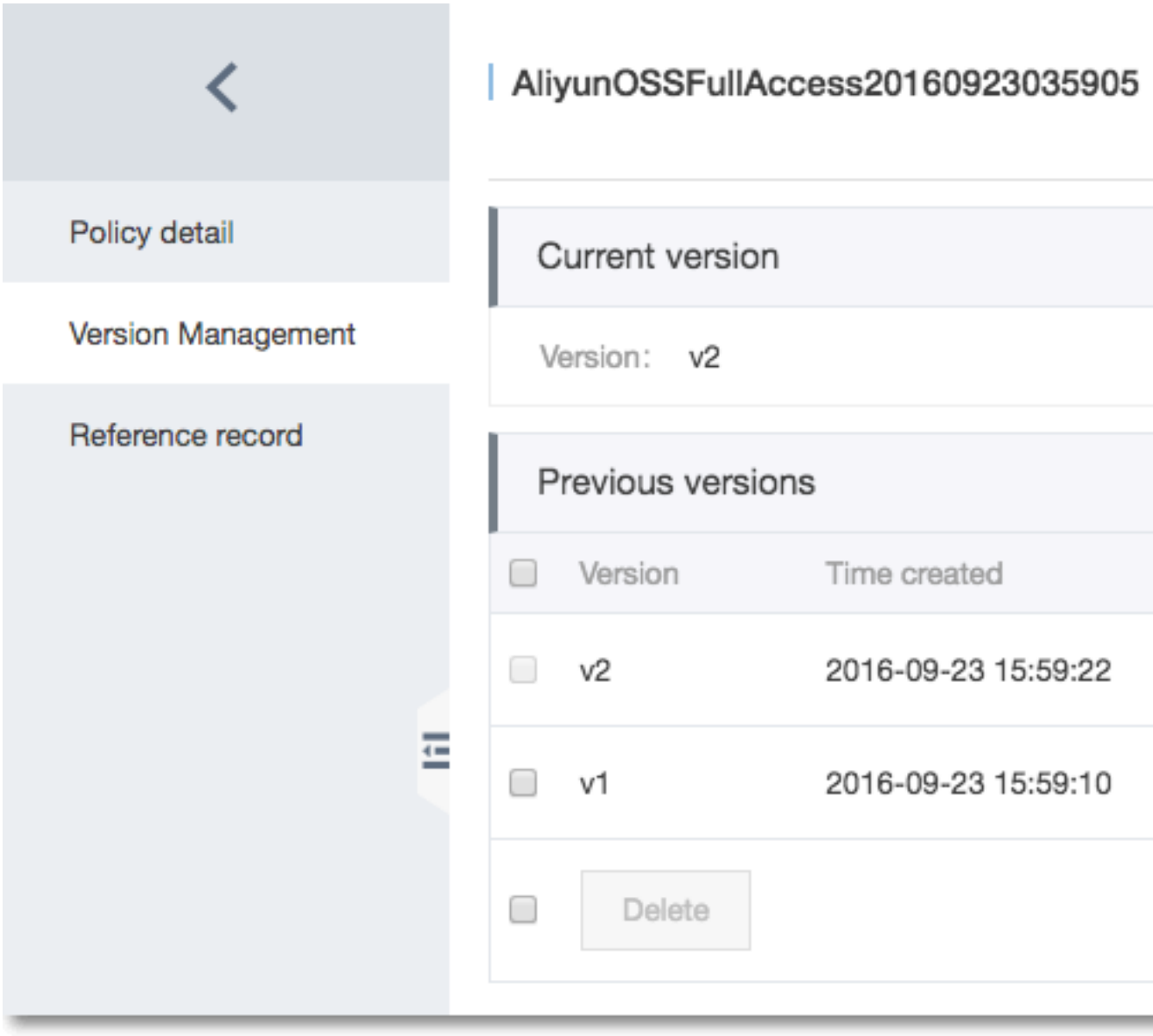
The authorization policy has a version management mechanism that addresses the problems that exist during use:

- You can retain multiple versions for one authorization policy.
- If beyond the limit, you need to delete the unnecessary versions.
- For an authorization policy having multiple versions, only one version is active, which is known as the "default version".

Operation steps

1. Click on the **Policy Management** > **Custom Authorization Policy**of the **RAM console**.
2. Through The Authorization Policy Name, which you can use to query by using the keyword , finds the authorization policy that needs to be managed, click the view whose name or corresponding action is listed.
3. In the left-hand navigation bar, click version management.

Figure 3-3: Version management



4. As shown in the figure above, on the version management page, you can:
   - Select to view policy content for all historical versions.
   - Sets the non-default version policy to the current version (that is, the default version).
   - Select remove non-default version policy.

Delete a custom authorization policy

You can create multiple authorization policies. Multiple versions can be maintained for each policy . Custom authorization policies no longer needed should be deleted.

Premise

Before you delete an authorization policy, you should ensure that:

- There are no multiple versions of the current authorization policy, and there is only one default version. If multiple versions of the authorization policy exist, you must first delete all versions except the default version.
- The current authorization policy is not referenced (that is, attached to a user, user group, or role). If the Authorization Policy has been referenced, you can:
  - Unauthorize in the reference record for the authorization policy.
  - You can also choose to force the relationship to be disassociated during the deletion process.

Operation steps

1. Click on the **Policy Management** > **Custom Authorization Policy** of the **RAM console**.
2. The Authorization Policy Name, which you can use to query by using the keyword, finds the authorization policy that needs to be deleted, click the delete on its corresponding action column.
3. Confirm the deletion of authorization policies, you can choose to check the force disassociate relationship (when the policy has been referenced records to force a reference relationship to be deleted).

So far, you have successfully deleted a custom Authorization Policy.

3.3 Authorization

In Ram, authorization is the process of attaching one or more authorization policies to a user, user group, or role. Wherein,

- Grant the user or user group authorization for the ram user under the current cloud account.
- Give Role authorization, both for the current cloud account, under the ram user license, also used on, other cloud accounts, under the ram user or service role Authorization; the difference is, the authorized object needs to act as a role to obtain the identity and permissions of the role.

Grant authorization to users or user groups

Ram under the current cloud account When users authorize, you can choose to authorize specific users, or you can authorize users to their user groups. The difference is that authorization to a user group applies to all users under the user group, users with similar requirements for resource access (created and added to the same group) conduct unified authorization.

Give the user authorization

The procedure is as follows:

1. Log on to the *RAM console*.
2. Click User management.
3. Users who require authorization can be found via user name/display name (fuzzy query can be used), click the authorization button that is listed in its corresponding action.
4. On the Edit personal authorization policy page,
   - From the optional Authorization Policy Name on the left, find the permissions that you want to grant to the current user (you can query using the keyword), check the policy and click the right arrow, you can add the policy to the right under the selected Authorization Policy Name.
   - Under Authorization Policy Name selected on the right, select a policy and click the left arrow, you can undo the policy.
5. After adding the Authorization Policy, click confirm to complete the authorization.

At this point, you have completed the authorization to the user.

Grant authorization to user groups

The procedure is as follows:

1. Log on to the *RAM console*.
2. Click group management.
3. The group name is used to locate the user groups that require authorization (Fuzzy queries are available), click the authorization that is listed in its corresponding action Button.
4. On Edit Group Authorization Policy, page
   - Find the permissions that need to be granted to the current group from the name of the optional Authorization Policy on the left (you can query using the keyword), check the policy and click the right arrow, you can add the policy to the right under the selected Authorization Policy Name.
   - Under Authorization Policy Name selected on the right, select a policy and click the left arrow, you can undo the policy.
5. After adding the Authorization Policy, click confirm to complete the authorization.

So far, you have completed authorization to the user group.

Give Role authorization

When you create a new role, you can choose to create a new user role (including using the current cloud account and other cloud accounts as trusted cloud accounts) or a service role, and you need to select the corresponding trusted cloud account or cloud service (that is, allow it to use the roles created. visit your cloud resources).

- For the current cloud account user role, authorization, then the ram users under the current cloud account can play a role and access the authorized cloud resources.
- Yes, other cloud account user roles, authorization, ram users under other designated cloud accounts can play a role and access authorized cloud resources..
- For service roles, authorization, A trusted cloud service can play a role and access authorized cloud resources.

Operation Steps

1. Log on to the *RAM console*.
2. Click role management.
3. Roles that require authorization are found by role name (fuzzy query can be used), click the authorization button that is listed in its corresponding action.
4. On the Edit Role authorization policy page,
   - From the left, the optional Authorization Policy Name The policy is selected by finding permissions that need to be granted to the current role (you can use the keyword query, and click the right arrow to add the policy to the right under the selected Authorization Policy Name.
   - Under Authorization Policy Name selected on the right, select a policy and click the left arrow, you can undo the policy.
5. After adding the Authorization Policy, click confirm to complete the authorization.

At this point, you have completed the authorization to the role.

3.4 Access resources

After authorization, RAM users can access the relevant resources through the console or the API. RAM users can also switch identities after logging on to the console, or call AssumeRole to obtain the role token (STS) to act as a relevant role and manipulate related resources as the role.

Access resources on the console

The RAM user logon requires an independent logon URL (which can be viewed on the RAM console). Use the **primary account enterprise alias**, **username**, and **password** to log on to the console. After successfully logging on, the user can perform operations on the authorized resources. If the user attempts to perform an operation that they do not have permission for, the error message “No operation permissions” is displayed.

If a RAM user is allowed to assume a role:

- After logon, the user can use the **Switch Role** operation to switch from the current logon identity to a role identity. In this way, the user can use the permissions of the newly selected role to perform operations on resources.
- If the user wants to switch back to the logon identity, the user can use the **Return to Logon Identity** operation.

For more information about roles, see *Role*.

Access resources from calling APIs

For the application that calls cloud service APIs to perform resource operations, you create a RAM user account for this application and grant it relevant permissions. Then, create an AccessKey for this RAM user, which is used by the application to call cloud service SDKs and APIs.

Access resources by using a client tool

Some cloud services provide easy-to-use client tools, for example, aliyuncli. These tools allow the usage of RAM user AccessKeys to perform cloud resource operations.

The following is an example of the OSS service. Assuming that the RAM user is authorized to access a bucket, you can then use the OSS client tool *ossbrowser* to access the specified bucket.

The operation procedure is as follows:

1. Open ossbrowser and set the account and password to AccessKeyId and AccessKeySecret, respectively.
2. After logging on, enter the ossbrowser interface. Select the **Authorized Bucket** tab and click **Add** to add an authorized bucket. Then, you can manipulate the content of the authorized bucket.

4 Policy Language

4.1 Elements

RAM uses an authorization policy to describe the content of authorization. The basic elements in an authorization policy include Effect, Resource, Action, and Condition.

Effect

Effects can be categorized into two types: Allow and Deny.

Resource

Resources are specific authorized objects.

For example, in the authorization policy “User A is allowed to perform the GetBucket operation on the resource SampleBucket”, the resource is “SampleBucket”.

Action

Actions are operations performed on specific resources.

For example, in the authorization policy “User A is allowed to perform the GetBucket operation on the resource SampleBucket”, the action is “GetBucket”.

Condition

Condition are the circumstances under which the authorization takes effect.

For example, in the authorization policy “User A is allowed to perform the GetBucket operation on the resource SampleBucket before 2011-12-31”, the condition is “before 2011-12-31”.

Example

This example authorization policy can be explained as follows: read-only operations on the OSS bucket samplebucket are allowed on the condition that the source IP address of the requester is 42.160.1.0.

```
      "Version": "1",
      "Statement":

          "Effect": "Allow",
            "Action": ["oss:List*", "oss:Get*"],
            "Resource": ["acs:oss:*:*:samplebucket", "acs:oss:*:*:
samplebucket/*"],
            "Condition":
```

```
                "IpAddress":
                    "acs:SourceIp": "42.160.1.0"
                  }
```

4.2 Policy syntax structure

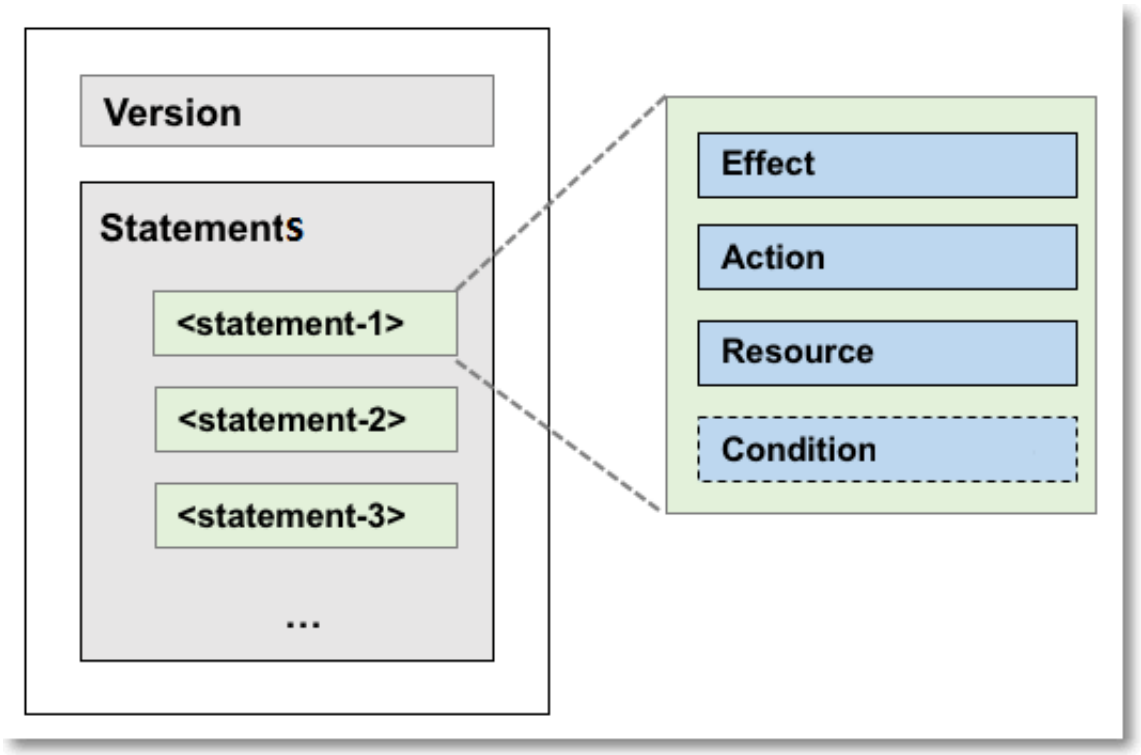
This article describes the syntax structure and rules of Authorization Policy in Ram, helps you understand and use it correctly. In everyday applications, you can quickly check out the following :, and in this article.

Policy Structure

Authorization Policy (policy) structure includes policy The version number and the list of authorized statements. Each authorization statement includes the following elements: Effect (authorization type), Action (action name list), resource (Action object list), and Condition, where condition is optional.

Basic policy structure:

Figure 4-1: Policy Structure



Format check (JSON)

Ram only supports descriptions in JSON format. When creating or updating a policy, RAM will first check that the JSON format is correct.

- For information on JSON syntax standards, refer to [RFC 7159].
- You can also use some online JSON format validators and editors to verify the validity of the JSON text.

Policy syntax

Understanding the characters and rules used in policy, and Policy The syntax description.

Characters and rules

The JSON characters contained in policy are: {} [] ", :; The special characters used in the description syntax are: = <> () |.

Instructions for use:

- When an element allows multiple values, it is expressed using a comma and ellipsis, such: [<action_string>, <action_string>, ...]. All syntaxes that support multiple values, also allow single values. And the two expressions are equivalent: "action ": [<Action_string>] and "action ": <Action_string>
- An element with a question mark indicates that this is an optional element, such as: <condition _block? >
- When multiple values are separated by a vertical bar (|), this indicates that only one of the values can be selected. For example: ("allow" | "deny ")
- An element enclosed with double quotes indicate a text string. For example: <version _block> = "version ": ("1 ")

Grammar description and description

Policy The syntax is described below:

```
Policy = {
                    <Version _block>,
                    <Wollongong _block>
}
<Version _block> = "version": ("1 ")
<Direct_block> = "statement": [<Statement>, <Statement>,...]
<Statement> = {
<Glast_block>,
<Action_block>,
<Think_block>,
<Condition _block? >
}
```

```
<Glast_block> = "effect": ("allow" | "deny ")
<Action_block> = ("action" | "notaction "):
("*" | [<Action_string>, <action_string>,...])
<Think_block> = ("resource" | "notresource "):
("*" | [<Think_string>, <think_string>,...])
<Condition _block> = "condition": <condition _map>
<Condition _map> = {
<Maid> :{
<Condition_key_string>: <condition_value_list>,
<Condition_key_string>: <condition_value_list>,
...
},
<Maid> :{
<Condition_key_string>: <condition_value_list>,
<Condition_key_string>: <condition_value_list>,
...
}, ...
}
<Condition_value_list> = [<condition _value>, <condition _value>,...]
<Condition _value> = ("string" | "Number" | "Boolean ")
```

The syntax is described below:

- Version: The currently supported version of policy is 1.
- Authorization statement: A policy can have multiple authorization statements.
  - * Each authorization statement can be either Deny or Allow. In an authorization statement , action is a list that supports multiple operations, Resource It is also a list that supports multiple objects.
  - Each authorization Statement supports separate conditions). A condition block supports multiple condition operation types and logical combinations of these conditions.
- Deny priority: a user can be granted multiple policies, follow these policies when there are multiple authorization statements that contain both allow and deny Deny first (only deny does not recognize allow) principle.
- Element value:
  - When the value is a number or Boolean, it is similar to a string, need to be caused by double quotation marks.
  - Supports (*) and (?) When element values are string values (?) Fuzzy Matching.
    - (*) Represents 0 or more arbitrary English letters.
    - (?) Represents 1 arbitrary Letter of English.

    For example, "ecs:Describe*" indicates all ECS API operation names starting with 'Describe '. Operation name.

The policy element uses

Understand the rules for the use of each element in the policy syntax.

Effect (authorization type)

The Effect value is either Allow or Deny. For example, "effect": "allow"

Action (operation name list)

Action supports multiple values, which is defined by the API operation name defined by the cloud service, and is defined in the following format:

```
<Service-Name>: <action-Name>
```

Description:

- * service-name: This indicates the name of an Alibaba Cloud product, such as ECS, RDS, Server Load Balancer, OSS, and Table Store.
- action-name: name of service-related action interfaces.

Description example:

```
"Action": ["Oss: listshoes", "ECs: Describe *", "RDS: Describe *"]
```

Resource (list of operation objects)

Resource usually refers to an operational object, such as ECS. Virtual Machine instance, OSS Store objects. Alibaba Cloud service resource names are formatted as follow:

```
ACS: <service-Name>: <region>: <account-ID>: <relative-ID>
```

Description:

- * acs: This is the abbreviation of Aliyun Cloud Service, indicating an Alibaba Cloud public cloud platform.
- * service-name: This indicates the name of an open service provided by Alibaba Cloud, such as ECS, OSS, or Table Store.
- * region: This indicates region information. If this option is not supported, use the wildcard "*" instead.
- Account-ID: account ID, such 1234567890123456 can also be replaced.
- * relative-id: This indicates the service-related resource. Its meaning is specified by the specific service. Specifies. This part of the format description supports a tree structure similar to the file path. In the case of OSS, relative-id = "Mybucket/dir1/object1.jpg" represents an OSS object.

Description example:

```
"Resource": ["ACS: ECs: *: instance/inst-001 ", "ACS: ECs: *: instance
/inst-002", "ACS: OSS: *: mybucket "," ACS: OSS :*:*: mybucket/* "]
```

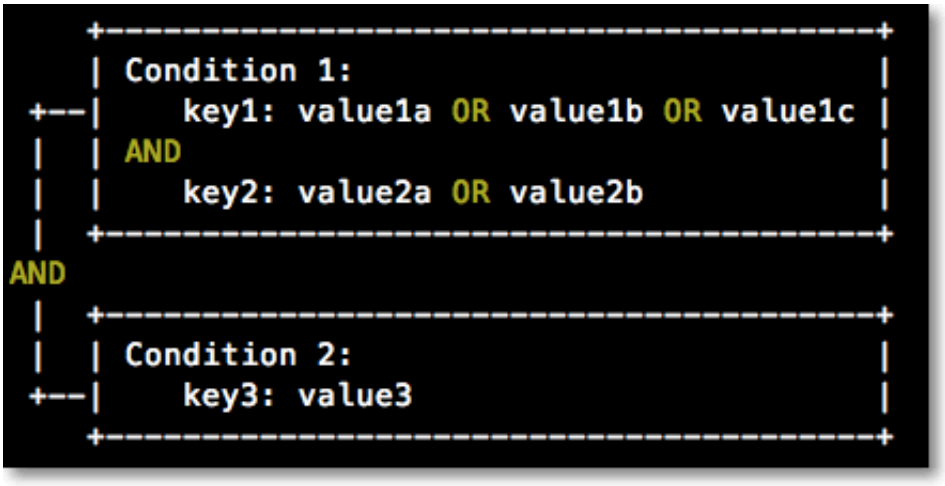
Condition (condition restrictions)

Condition block (Condition Block) consists of one or more conditional clauses. A condition clause consists of an action type, keyword, and condition value. The action type and keyword are to be elaborated in the following sections.

Conditional block judgment Logic

The following figure shows the criteria for determining whether a condition is met.

Figure 4-2: Are the judgment principles that meet the conditions?



The specific rules are as follows:

- One condition keyword corresponds to one or more values. When conditions are checked, if the condition keyword value is equal to one of the corresponding values, the condition is considered satisfied.
- A condition clause is considered satisfied only if multiple condition keywords of the condition clause of the same condition action type are all satisfied.
- A condition block is satisfied only if all its condition clauses are satisfied.

Condition action type

Supports the following conditions: string type, numeric type), date type (data and Time, Boolean, and IP address type (IP address).

These condition action types support the following methods, respectively:

String	Numeric	Date and time	Boolean	IP address
Stringequals	Numericequals	Dateequals	bool	IPaddress
Stringnotes	Numericnot equals	Datenotequals	-	Notipaddress
Stringequa lsignorecase	Numericlessthan	Datelessthan	-	-
Stringnote qualsignorecase	Numericles sthanequals	Datelessth anequals	-	-
Stringlike	Numericgre aterthan	Dategreaterthan	-	-
Stringnotlike	Numericgre aterthanequals	Dategreate rthanequals	-	-

Conditions keyword (Condition-key)

The condition keywords reserved by Alibaba Cloud adopt the following naming format:

```
ACS: <condition-key>
```

The common condition keywords reserved by the Ali Cloud Service are as follows:

General Condition keyword	Type	Description
`acs: CurrentTime`	Date and time	The time that the Web server received the request in ISO 8601 format, such `2012-11-11T23: 59: 59Z`
`acs: SecureTransport`	Boolean	Whether the request is sent over a secure channel, such as via HTTPS.
`acs: SourceIp`	IP address	The IP address of the console sending the request.
`ACS: mfapresent`	Boolean	Whether multi-factor authentication is used during user login (two-step authentication)

Some products define product-level condition keywords, in the following format:

```
<Service-Name>: <condition-key>
```

Some cloud products define conditional keywords as follows:

Product Name	Conditional keyword	Type	Description:
ECS	`ECS: tag/<tag-key>`	String	Tag keywords for ECs resources that can be customized by the user
RDS	`RDS: thinktag/<tag-key>`	String	Tag keyword for the RDS resource that can be customized by the user
OSS	`OSS: impression iter`	String	The separator that the OSS groups object names
	`OSS: prefix`	String	Prefix of the OSS Object Name

Policy sample

The policy below contains two authorization statements.

- 1st authorization statements are allowed on Region East 1 (Hangzhou) All ECs Resources have permission to view (ECS: Describe *);
- 2nd authorization statement is to allow read access to objects in the mybucket bucket of OSS (OSS: listObjects, OSS: GetObject), and limits the IP of the requestor Source must be 42.120. 88.10 or 42.120.66.0/24.

```
{
    Version: "1 ",
    "Statement ":[
        {
            "Effect": "allow ",
            "Action": "ECs: Describe *",
            "Resource": "ACS: ECs: CN-Hangzhou :*:*"
        },
        {
            "Effect": "allow ",
            "Action ":[
                "Oss: maid ",
                "Oss: GetObject"
            ],
            "Resource ":[
                "ACS: OSS: *: mybucket ",
```

```
                "ACS: OSS: *: mybucket /*"
            ],
            "Condition ":{
                "IPaddress ":{
                    "ACS: sourceip": ["maid", "maid/24"]
                }
            }
        }
    ]
}
```

4.3 Authorization rules

To help you better understand the authorization policy, this document describes the authorization model and rules.

Basic model

When a user attempts to access a resource by using different identities, RAM evaluates the access with corresponding logic.

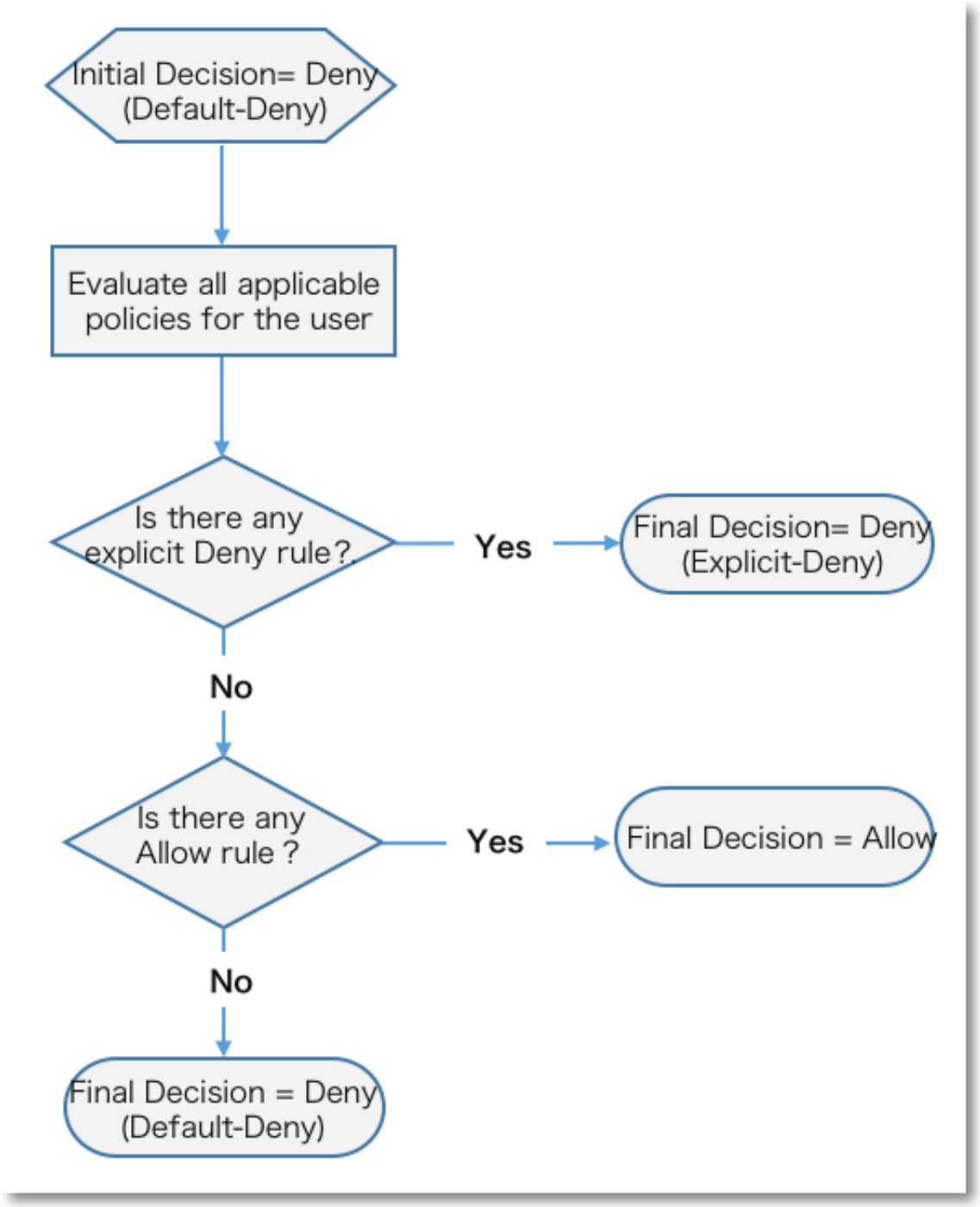
Access From	The Access is Permitted Only When
A primary account	The user is the resource owner. **Exception** : A few cloud products, such as SLS, directly support cross-account ACL authorization. If the ACL authorization check is passed, access is permitted.
A RAM-User identity	• The primary account to which the RAM-User belongs has the access permission for the resource. • The primary account has attached an explicit Allow authorization policy to the RAM-User.
A RAM-Role identity	• The primary account to which the RAM-Role belongs has the access permission for the resource. • The primary account has attached an explicit Allow authorization policy to the RAM-Role. • The role’s STS-Token is explicitly authorized .

Authorization policy check logic for a RAM-User identity

By default, RAM-Users do not have resource access permissions unless they have been explicitly authorized by the primary account (that is, they have been attached with an authorization policy). Authorization policy statements support two types of authorization: Allow and Deny. When multiple authorization policy statements grant Allow and Deny permissions for the same resource, **Deny** takes priority.

The following figure details the authorization policy check logic.

Figure 4-3: Authorization policy check logic



When a RAM user accesses resources, the authorization policy check logic is as follows:

1. The system checks whether a RAM-User identity is authorized according to the authorization policy attached with the RAM-User identity.
   - If the result is Deny, access is denied.
   - Otherwise, the system proceeds with the next stage.

2. The system checks whether the primary account of the RAM-User has access permission for the resource.
   - If the account is the resource owner, access is permitted.
   - If the account is not the resource owner, the system checks whether the resource supports cross-account ACL authorization.
     - If yes, access is permitted.
     - Otherwise, access is denied.

Authorization policy check logic for a RAM-Role identity

If a user attempts to access a resource by using a RAM-Role (that is, using an STS-Token), the authorization policy check logic is as follows:

1. If the current STS-Token specifies an authorization policy (the authorization policy parameters entered when the AssumeRole API is called), the authorization policy check logic described in the preceding section is implemented.
   - If the result is Deny, access is denied.
   - Otherwise, the system proceeds with the next stage.

   If the STS-Token does not specify an authorization policy, the system automatically goes to the next stage.
2. The system checks whether the RAM-Role identity is authorized according to the authorization policy attached with the RAM-Role identity.
   - If the result is Deny, access is denied.
   - Otherwise, the system proceeds with the next stage.
3. The system checks whether the primary account of the RAM-User has access permission for the resource.
   - If the account is the resource owner, access is permitted.
   - If the account is not the resource owner, the system checks whether the resource supports cross-account ACL authorization.
     - If yes, access is permitted.
     - Otherwise, access is denied.

5 Cloud product authorization

6 Scenarios

6.1 RAM user management and authorization for enterprises

Scenario description

Assume that enterprise A buys several types of cloud resources, such as ECS instances, RDS instances, SLB instances, and OSS buckets. Employees at enterprise A need to perform operations on these resources, such as procurement, O&M, or online application. Because different employees have different responsibilities, they require different permissions.

- For security reasons, the Alibaba Cloud account owner of enterprise A does not want to disclose its account AccessKey to its employees. Rather, the account owner prefers to create different RAM user accounts for their employees and associate each RAM user account with different permissions.
- Then, the employees can perform resource operations only under their permissions with their RAM user accounts and charges are not billed to these accounts. All expenses are charged to the account owner.
- The account owner can also revoke the permissions of a RAM user account at any time, and delete the user.

Requirement analysis

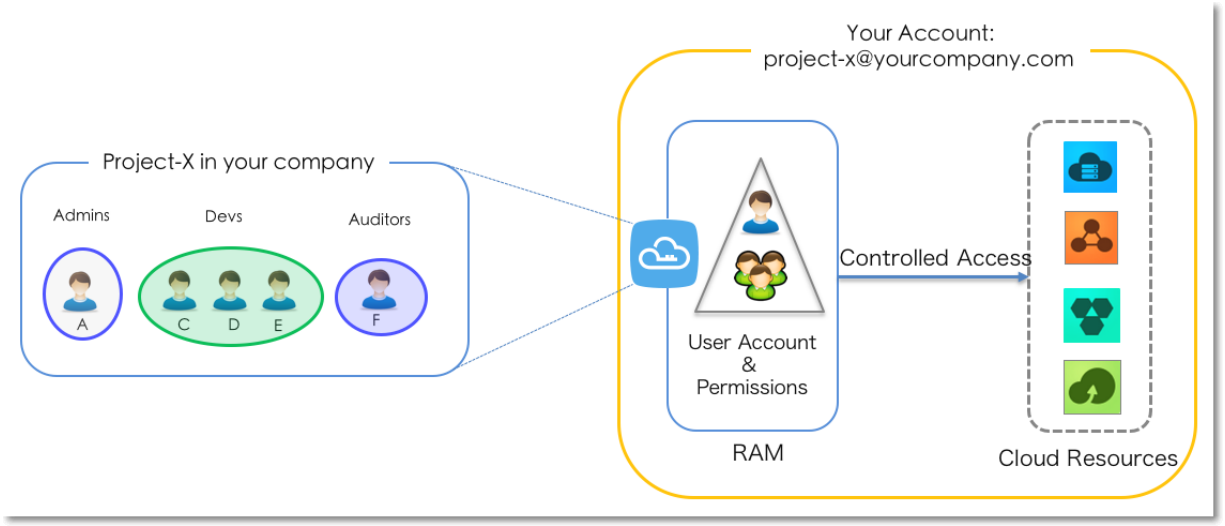
The analysis of the preceding scenarios is as follows:

- Employees do not share the primary account to avoid uncontrollable risks caused by the disclosure of the account’s password or AccessKey.
- Different employees are allocated independent user accounts (or operator accounts) with independent permissions, so that their responsibilities are consistent with their permissions.
- All the operations of all user accounts can be audited.
- Charges are not calculated for each operator; the primary account is billed for all fees incurred.

Solution

Use RAM-user accounts and the authorization management function, as shown in the following figure:

Figure 6-1: Solution for the enterprise scenario



The operation procedure is as follows:

1. *Set up MFA* to prevent risks caused by disclosure of the password of the primary account.
2. Activate RAM.
3. *Create RAM users* for different employees (or application systems) and set logon passwords or create AccessKeys for them as needed.
4. *Create a RAM user group*. If multiple employees share the same responsibility, we recommend that you create a group for them and add the users to the group.
5. *Authorization*. Attach one or more authorization policies to groups or users. For finer-grained authorization, you can create *Authorization Policy Management* and then attach them to groups or users.

6.2 Temporary authorization management of mobile apps

This document describes how to use the RAM role token to control temporary authorization access for mobile apps in specific scenarios.

Scenario description

Assume that enterprise A has developed a mobile app and has bought OSS for it. The mobile app must upload and download data to and from OSS. Because the mobile app runs on user devices, these devices are out of A's control.

- Enterprise A does not want to allow all apps to use the appServer to transmit data. Instead, enterprise A wants the apps to directly upload and download data to and from OSS.
- For security reasons, enterprise A cannot save the AccessKey in the app.

- Enterprise A also wants to minimize its security risks by, for example, giving each app an access token with the minimum permissions that the app needs to connect to OSS and restricting the access duration to a specified period of time (such as 30 minutes).

Requirement analysis

The analysis of the preceding scenarios is as follows:

- The mobile app needs to directly transmit data to OSS, without using a data proxy.
- Enterprise A cannot give an AccessKey to the mobile app because the mobile devices are under the control of A's users.
- The access permissions of each mobile app must be restricted to OSS object granularity.

Solution

In response to the above requirements, **the RAM role token is used to authorize temporary access to the OSS**.

- Account A creates a role in RAM and assigns appropriate permissions to the role, and allows the appServer (logs on as the RAM user) to use this role. For the operation procedure, see the section "Create roles, users, and authorizations".
- When the app needs to upload or download data to OSS through direct connection, the appServer can assume a role (calling STS AssumeRole) to get a temporary security token (STS-Token) for the role and transfer it to the app. Then, the app can use the temporary security token to access the OSS API directly. For the operation procedure, see the section " Get and pass the role token".
- The appServer can further limit the resource operation permissions of the temporary security token while using the role, to control the permissions of each app in greater detail. For the operation procedure, see the section "Restrict the STS-Token permission".

Create roles, users, and authorizations

Assume that account A's account D is: 11223344. The process for creating roles, users, and configuring permissions for the appServer is as follows:

1. Account A creates a user role (assuming the role is named oss-readonly) and selects the **Current Alibaba Cloud Account** as the trusted account. That is, only the RAM users under account A are allowed to assume this role. For the operation procedure, see *Role*.

   After creating the role, enterprise A can get the role information on the details page.

- In this example, the role's global name ARN is:

```
acs:ram::11223344:role/oss-readonly
```

- The role's policy (only the RAM users under account A can assume this role) is as follows:

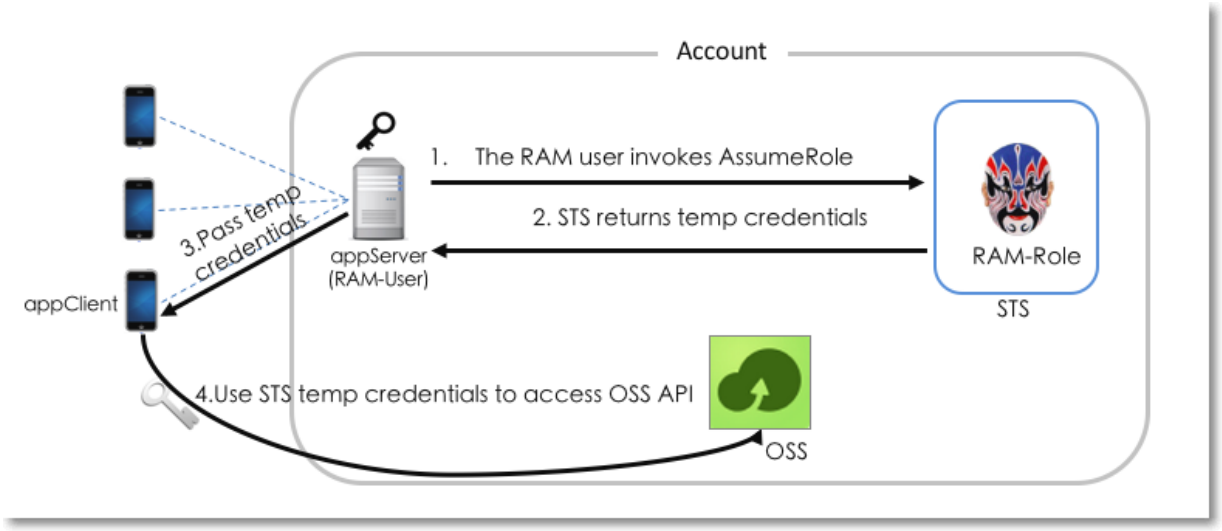
```
{
"Statement": [
{
 "Action": "sts:AssumeRole",
 "Effect": "Allow",
 "Principal": {
   "RAM": [
     "acs:ram::11223344:root"
   ]
 }
}
],
"Version": "1"
}
```

2. Account A adds the policy AliyunOSSReadOnlyAccess to the role oss-readonly.
3. Account A creates a RAM user role for the appServer (assuming the user name is appServer) and:
   - creates an AccessKey for the RAM user. That is, the RAM user appServer is allowed to call the API.
   - calls the permissions for the STS interface AssumeRole (AliyunSTSAssumeRoleAccess). That is, the RAM user appServer is allowed to assume the role.

Get and pass the role token

The procedure for the appClient to get and use the role token to call the OSS API is as follows:

Figure 6-2: Operation procedure



The operation procedure is as follows:

1. The appServer uses the AccessKey of the RAM user (appserver) to call STS AssumeRole. The command example of using aliyuncli to call AssumeRole is as follows:

> **Note:**
> The AccessKey for the appServer must be configured, and the AccessKey for primary account A is not allowed.

```
$ aliyuncli sts AssumeRole --RoleArn acs:ram::11223344:role/oss-
readonly --RoleSessionName client-001
 {
     "AssumedRoleUser": {
         "AssumedRoleId": "391578752573972854:client-001",
         "Arn": "acs:ram::11223344:role/oss-readonly/client-001"
     },
     "Credentials": {
         "AccessKeySecret": "93ci2umK1QKNEja6WGqi1Ba7Q2Fv9P
wxZqtVF2VynUvz",
         "SecurityToken": "CAES6AIIARKAAUiwSHpkD3GXRMQk9stDr3YSVbyG
qanqkS+fPlEEkjZ+dlgFnGdCI2PV93jksole8ijH8dHJrHRA5JA1YCGsfX5hrzcNM3
7Vr4eVdWFVQhoCw0DXBpHv//ZcITp+ELRr4MHsnyGiErnDsXLkI7q/sbuWg6PACZ
/jzQfEWQb/f7Y1Gh1TVFMuRjEzR2pza1hUamszOGRCWTZZeEp0WEFaayISMz
kxNTc4NzUyNTczOTcyODU0KgpjbGllbnQtMDAxMKT+lIHBKjoGUnNhTUQ1QkoK
ATEaRQoFQWxsb3cSGwoMQWN0aW9uRXF1YWxzEgZBY3Rpb24aAwoBKhIfCg5S
ZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaAwoBKkoFNDMyNzRSBTI2ODQyWg9B
c3N1bWVkUm9sZVVzZXJgAGoSMzkxNTc4NzUyNTczOTcyODU0cgllY3MtYWRt
aW544Mbewo/26AE=",
         "Expiration": "2016-01-13T15:02:37Z",
         "AccessKeyId": "STS.F13GjskXTjk38dBY6YxJtXAZk"
     },
     "RequestId": "E1779AAB-E7AF-47D6-A9A4-53128708B6CE"
```

```
}
```

Restrict the STS-Token permission

- The Policy parameter is not specified when the AssumeRole above is called, indicating that the STS-Token has all permissions of oss-readonly.
- If you need to further restrict the permission of STS-Token, for example, only access to `sample-bucket/2015/01/01/*.jpg` is allowed, you can restrict the permission of STS-Token in greater detail through the Policy parameter. For example:

```
$ aliyuncli sts AssumeRole --RoleArn acs:ram::11223344:role/oss
-readonly --RoleSessionName client-002 --Policy "{\"Version\":
\"1\", \"Statement\": [{\"Effect\":\"Allow\", \"Action\":\"oss:
GetObject\", \"Resource\":\"acs:oss:*:*:sample-bucket/2015/01/01/
*.jpg\"}]}"
{
   "AssumedRoleUser": {
       "AssumedRoleId": "391578752573972854:client-002",
       "Arn": "acs:ram::11223344:role/oss-readonly/client-002"
   },
   "Credentials": {
       "AccessKeySecret": "28Co5Vyx2XhtTqj3RJgdud4ntyzrSN
dUvNygAj7xEMow",
       "SecurityToken": "CAESnQMIARKAASJgnzMzlXVyJn4KI+FsysaIpTGm
8ns8Y74HVEj0pOevO8ZWXrnnkz4a4rBEPBAdFkh3197GUsprujsiU78Fkszx
hnQPKkQKcyvPihoXqKvuukrQ/Uoudk31KAJEz5o2EjlNUREcxWjRDRSISMzkxNTc4
NzUyNTczOTcyODU0KgpjbGllbnQtMDAxMKmZxIHBKjoGUnNhTUQ1Qn8KATEa
egoFQWxsb3cSJwoMQWN0aW9uRXF1YWxzEgZBY3Rpb24aDwoNb3NzOkdldE9i
amVjdBJICg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaLAoqYWNzOm9zczoq
Oio6c2FtcGxlLWJ1Y2tldC8yMDE1LzAxLzAxLyouanBnSgU0MzI3NFIFMjY4
NDJaD0Fzc3VtZWRSb2xlVXNlcmAAahIzOTE1Nzg3NTI1NzM5NzI4NTRyCWVj
cy1hZG1pbnjgxt7Cj/boAQ==",
       "Expiration": "2016-01-13T15:03:39Z",
       "AccessKeyId": "STS.FJ6EMcS1JLZgAcBJSTDG1Z4CE"
   },
   "RequestId": "98835D9B-86E5-4BB5-A6DF-9D3156ABA567"
}
```

- In addition, the default validity period of the above STS-Token is 3600 seconds. You can use the DurationSeconds parameter to limit the STS-Token expiration time (it cannot exceed 3600 seconds).

2. The appServer retrieves and parses the credentials.

- The appServer retrieves the AccessKeyId, AccessKeySecret and SecurityToken from the credentials returned by AssumeRole.
- Because the STS-Token validity period is relatively short, if the application requires a longer validity period, the appServer must re-issue a new STS-Token (for example, issue one STS-Token every other 1800 seconds).

3. The appServer securely transmits an STS-Token to the appClient.

4. The appClient uses the STS-Token to directly access a cloud service API (such as OSS). The operation commands for aliyuncli to use an STS-Token to access an OSS object are as follows (an STS-Token is issued to client-002):

```
Configure STS-Token syntax: aliyuncli oss Config --host --accessid
 --accesskey --sts_token
$ aliyuncli oss Config --host oss.aliyuncs.com --accessid STS.
FJ6EMcS1JLZgAcBJSTDG1Z4CE --accesskey 28Co5Vyx2XhtTqj3RJgdud4ntyzrSN
dUvNygAj7xEMow --sts_token CAESnQMIARKAASJgnzMzlXVyJn4KI+FsysaIpTGm
8ns8Y74HVEj0pOevO8ZWXrnnkz4a4rBEPBAdFkh3197GUsprujsiU78Fkszx
hnQPKkQKcyvPihoXqKvuukrQ/Uoudk31KAJEz5o2EjlNUREcxWjRDRSISMzkxNTc4
NzUyNTczOTcyODU0KgpjbGllbnQtMDAxMKmZxIHBKjoGUnNhTUQ1Qn8KATEa
egoFQWxsb3cSJwoMQWN0aW9uRXF1YWxzEgZBY3Rpb24aDwoNb3NzOkdldE9i
amVjdBJICg5SZXNvdXJjZUVxdWFscxIIUmVzb3VyY2UaLAoqYWNzOm9zczoq
Oio6c2FtcGxlLWJ1Y2tldC8yMDE1LzAxLzAxLyouanBnSgU0MzI3NFIFMjY4
NDJaD0Fzc3VtZWRSb2xlVXNlcmAAahIzOTE1Nzg3NTI1NzM5NzI4NTRyCWVj
cy1hZG1pbnjgxt7Cj/boAQ==
Access OSS objects
$ aliyuncli oss Get oss://sample-bucket/2015/01/01/grass.jpg grass.
jpg
```

More references

More references to mobile app access include the following topics in OSS documentation:

- Set up direct data transfer for mobile apps
- Construct an STS policy for an app server
- Set up data callback for mobile apps
- STS temporary access authorization

6.3 Cross-account resource access and authorization

This document describes how to use RAM roles to perform cross-account resource access and authorization in specific scenarios.

Scenario description

Account A and account B represent different enterprises (teams or projects). Assume that enterprise A has bought a lot of cloud resources, such as ECS instances, RDS instances, SLB instances, and OSS buckets for its business requirements.

- Enterprise A wants to focus on its business systems, so it grants cloud resource O&M, monitoring management, and other tasks to the enterprise B.
- Enterprise B can then further delegate O&M tasks to its employees. Enterprise B needs to precisely control the delegate operations that its employees can perform on the cloud resources of the enterprise A.

- If A and B terminate this O&M entrustment contract, enterprise A is able to revoke the permissions of the enterprise B as needed.

Requirement analysis

The analysis of the preceding scenarios is as follows:

- Authorization between two Alibaba Cloud accounts, A and B. Account A is the resource owner and wants to grant B permissions to perform operations on its resources.
- Account B needs to further allocate permissions to its sub-users (employees or applications). If an employee of B joins or leaves the company, A does not have to make any changes to the permissions.
- If A and B terminate their cooperation, A is able to revoke B’s permissions as needed.

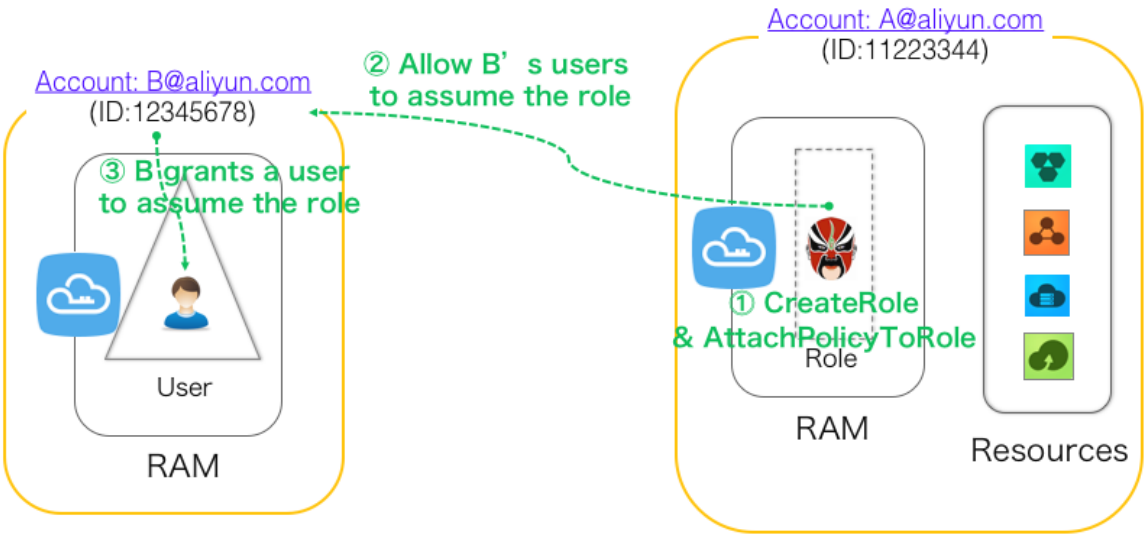
Solution

For the requirements above, **use the RAM role for cross-account authorization and resource access**.

- Account A creates a role in RAM and assigns appropriate permissions to the role, and allows account B to use this role. For the operation procedure, see the section "Cross-account authorization".
- If an employee (RAM user) under account B needs to use this role, account B can perform authorization control independently. Under the O&M entrustment contract, the RAM user under account B can use the granted role identity to manipulate the resources of account A. For the operation procedure, see the section "Cross-account resource access".
- If A and B terminate this O&M entrustment contract, account A only needs to revoke the enterprise B's use of this role. Once account B's use of this role is revoked, all RAM users under account B cannot use this role. For the operation procedure, see the section "Revocation of cross-account authorization".

Cross-account authorization

The following figure shows how to use the RAM role for cross-account authorization. Assume that enterprise A (AccountID=11223344, alias: company-a) needs to grant ECS operation permissions to the employees of enterprise B (AccountID=12345678, alias: company-b).



The operation procedure is as follows:

1. Account A creates a user role (assuming the role is named ecs-admin) and selects **Other Alibaba Cloud Account** (account B: 12345678). As a trusted account, the RAM users under account B are allowed to assume this role. For the detailed operation, see *Role*.

   After creating the role, enterprise A can get the role information on the details page.

   - In this example, the role's global name ARN is:

     ```
     acs:ram::11223344:role/ecs-admin
     ```

   - The role's policy (only enterprise B can assume this role) is as follows:

     ```
     "Statement": [

      "Action": "sts:AssumeRole",
      "Effect": "Allow",
      "Principal": {
        "RAM": [
          "acs:ram::12345678:root"

      }


     "Version": "1"
     ```

2. Account A *adds the authorization policy* (AliyunECSFullAccess) to the role ecs-admin.
3. Account B creates a RAM user for its employee (assuming the user name is Alice) and

   - *sets the logon password* (assuming the logon password is 123456). That is, the RAM user is allowed to log on to the console.

- *calls the permission for the STS interface AssumeRole* (AliyunSTSAssumeRoleAccess). That is, the RAM user Alice is allowed to assume or switch the role.

Cross-account resource access

The RAM user Alice under account B accesses account A's ECS resources through the console. The operation procedure is as follows:

1. The RAM user (Alice) under account B *logs on to the console*.

   When a sub-user logs on, the sub-user must enter **enterprise alias** (company-b), **sub-user username** (Alice), and **sub-user password** (123456) correctly.
2. The RAM user (Alice) under account B *switches the role*.

   In the upper right corner of the console, move the mouse pointer to the logon user name and click **Switch Role** to go to the identity switching page. Enter **enterprise alias** (company-a) and **role name** (ecs-admin) correctly to **switch** the role.
3. The RAM user (Alice) under account B manipulates the ECS resources under account A.

Revocation of cross-account authorization

Account A revokes account B's use of the role ecs-admin. The operation procedure is as follows:

1. Account A logs on to the RAM console, finds the role ecs-admin on the **Role Management** page, and clicks the name of its role or the **Manage** button to go to the **Role Details** page.
2. Click **Edit Basic Information** in the upper right corner. In the displayed dialog box, remove `acs:ram::12345678:root` from **Policy Content**. (That is, account B will be removed from the trusted cloud accounts of the role.)

   Note:
   Alternatively, account A can **Delete** the role ecs-admin from the **Role Management** page. Before the deletion, ensure that the role does not have any authorization policies.

7 RAM operation records

7.1 Use ActionTrail to log RAM operations

RAM has been integrated with *ActionTrail console*, with which you can view the operation and maintenance records of all users (the primary account and RAM users) on your instances.

ActionTrail logs the following RAM information:

- Logon of the primary account and RAM users. For more information, see *ConsoleSignin ConsoleSignin event log examples*.
- RAM console operations. For more information, see **A RAM user uses the RAM service in the console** in *RAM event log examples*.
- RAM/STS API calling records concerning resource creation, change, or deletion. For more information, see **A RAM user uses the RAM service through the SDK** in *RAM event log examples*.

For operation record details, see *ActionTrail event log syntax*.

8 Google Authenticator installation and user guide

8.1 Google Authenticator installation and user guide

Google Authenticator is a software token that implements two-step verification services using the TOTP (*RFC 6238*), for authenticating users of mobile applications by Google.

Select your operating system

- *iOS-based Google Authenticator installation and use guide*
- *Android-based Google Authenticator installation and use guide*

Note:
Because the token is time-based, make sure that the time on your device is accurate.

8.2 iOS-based Google Authenticator installation and use guide

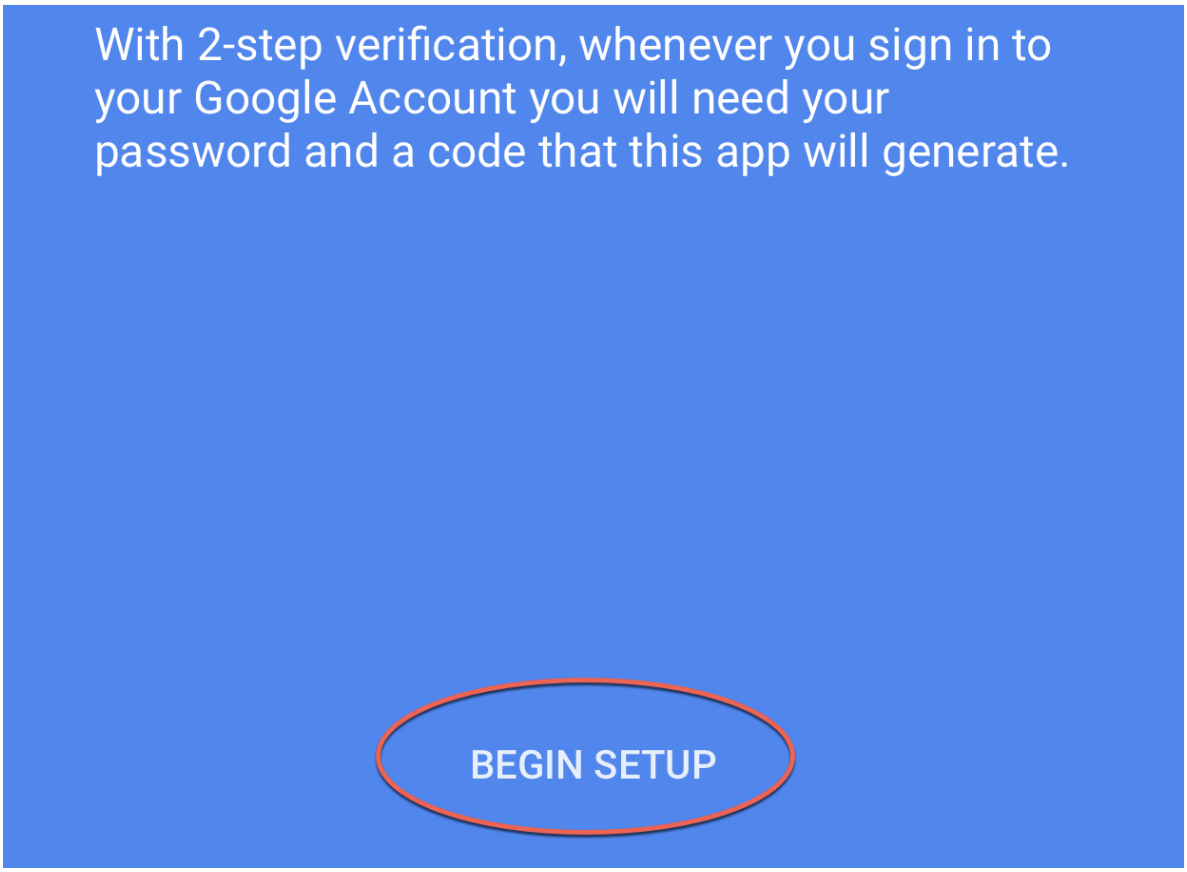
Installation

You can go to the Apple App Store and search for “Google Authenticator” to install the app.

Configuration

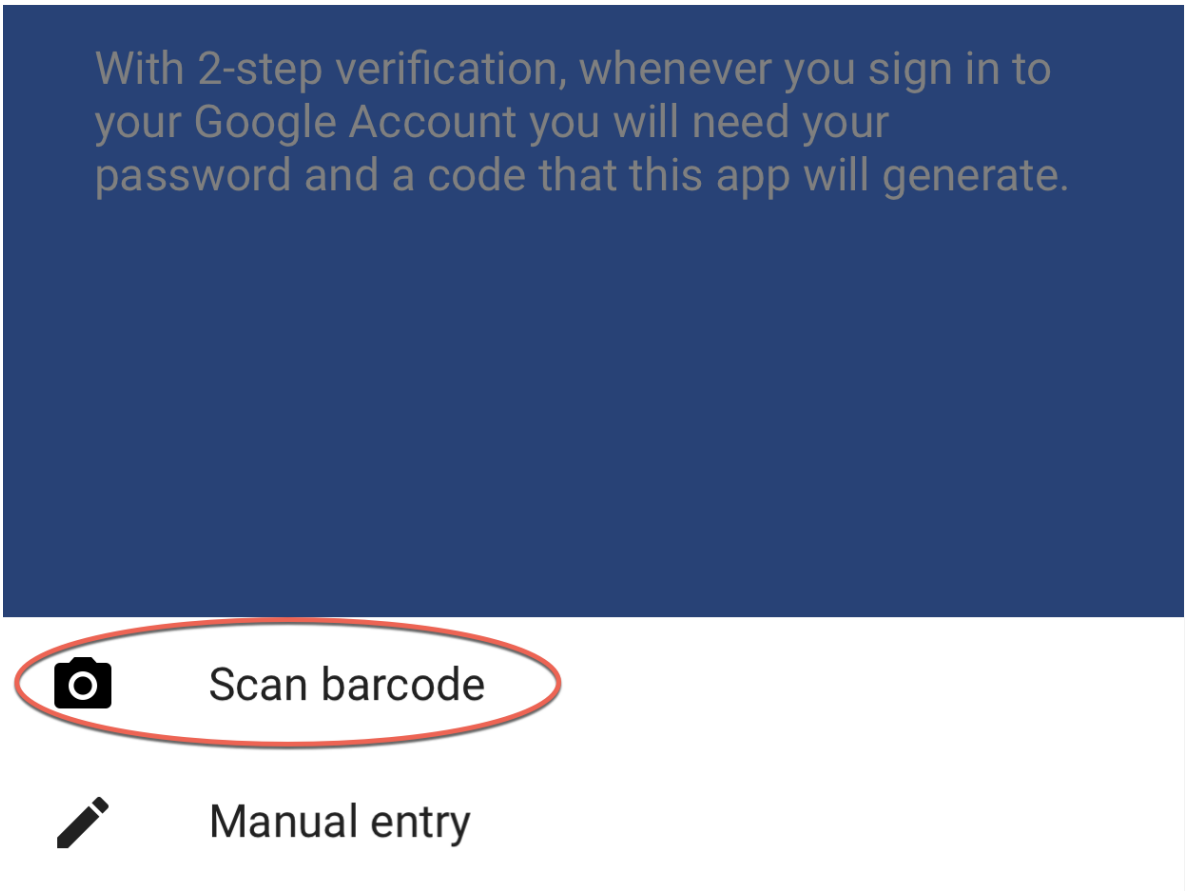
1. Open Google Authenticator and click **BEGIN SETUP** at the bottom of the page.

Figure 8-1: BEGIN SETUP



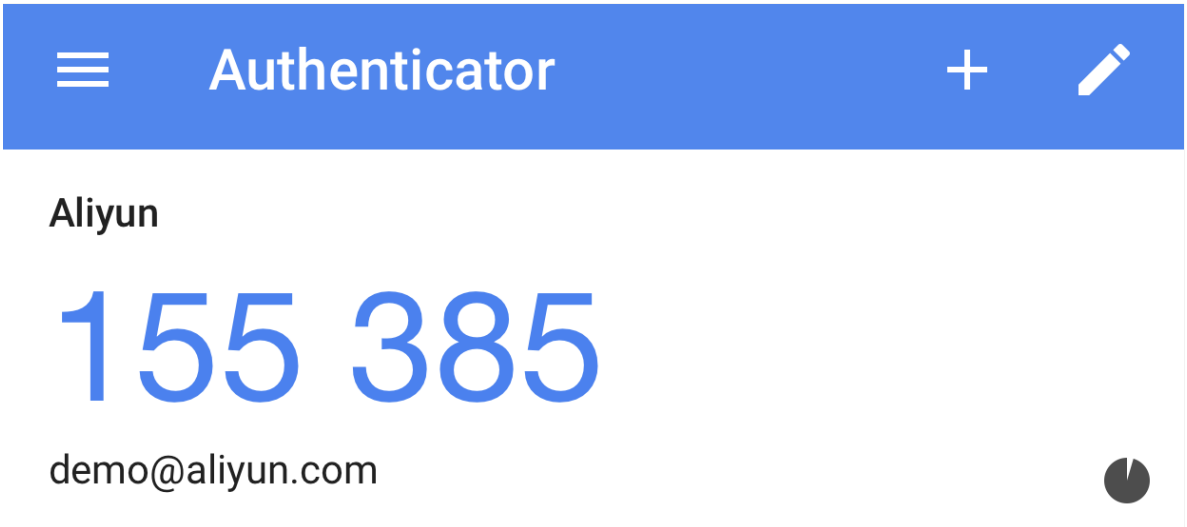
2. Select **Scan barcode** and then scan the barcode generated on the MFA binding page.

Figure 8-2: Scan barcode



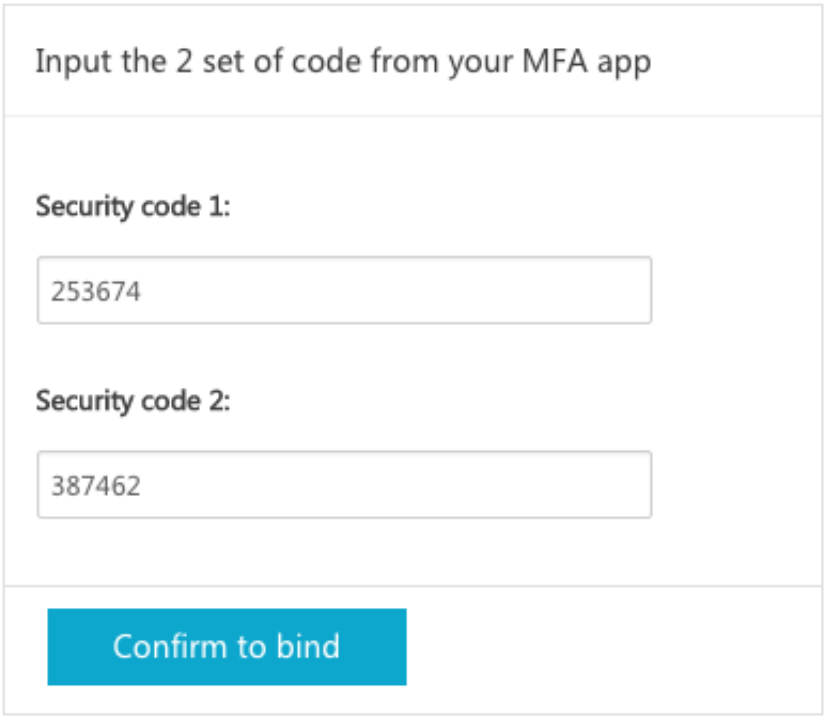
3. After scanning the code, you will see the window as shown in the following figure that displays your account name and MFA key.

Figure 8-3: Authenticator



4. On the MFA page, enter the two consecutive MFA codes and then click **Confirm to bind** to bind the authenticator.

Figure 8-4: Confirm to bind



8.3 Android-based Google Authenticator installation and use guide

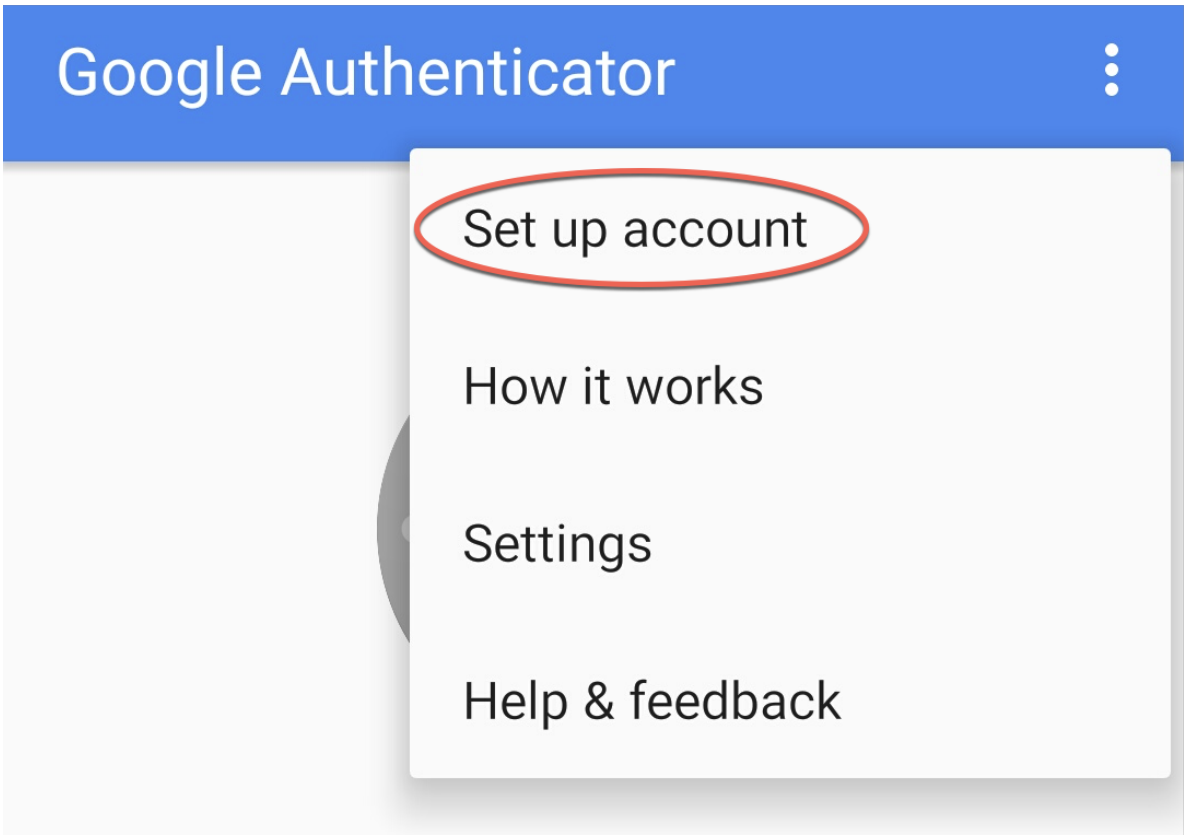
Installation

Google Authenticator is an Android-based application that implements two-step verification services. You can search for “Google Authenticator" in the Google Play Store, or a supporting app market, to install this app.

Configuration

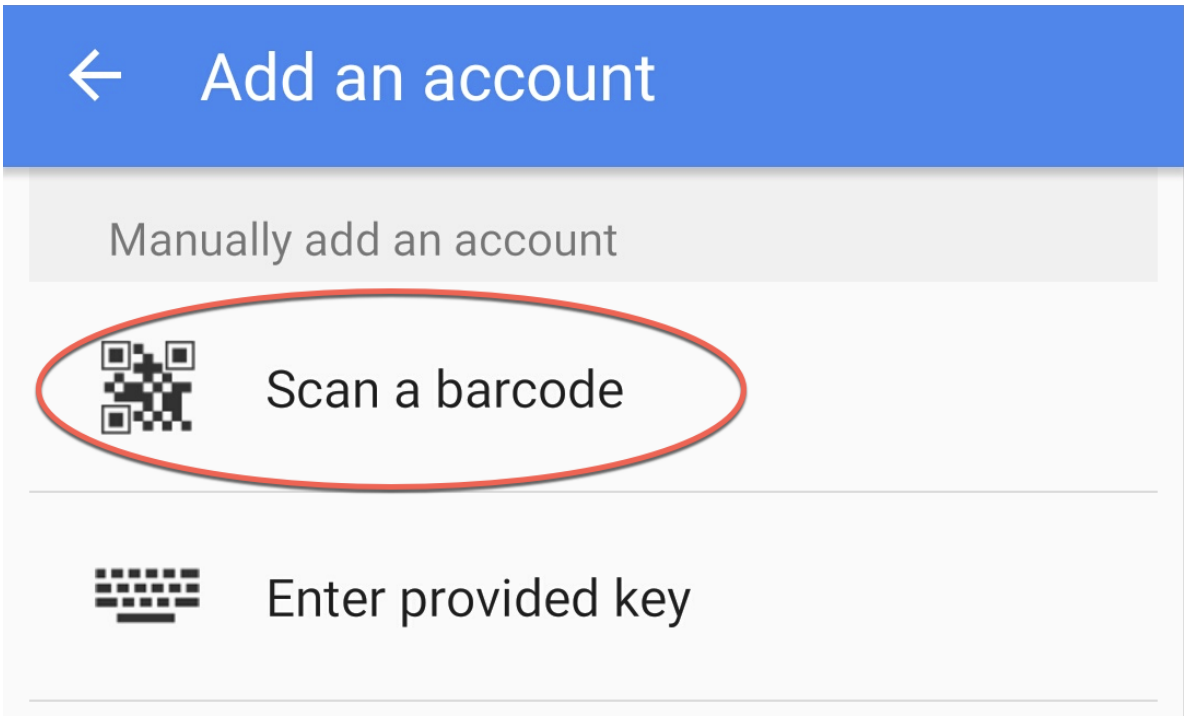
1. Open Google Authenticator and select the **Set up account** option in the menu in the top-right corner.

Figure 8-5: Set up account



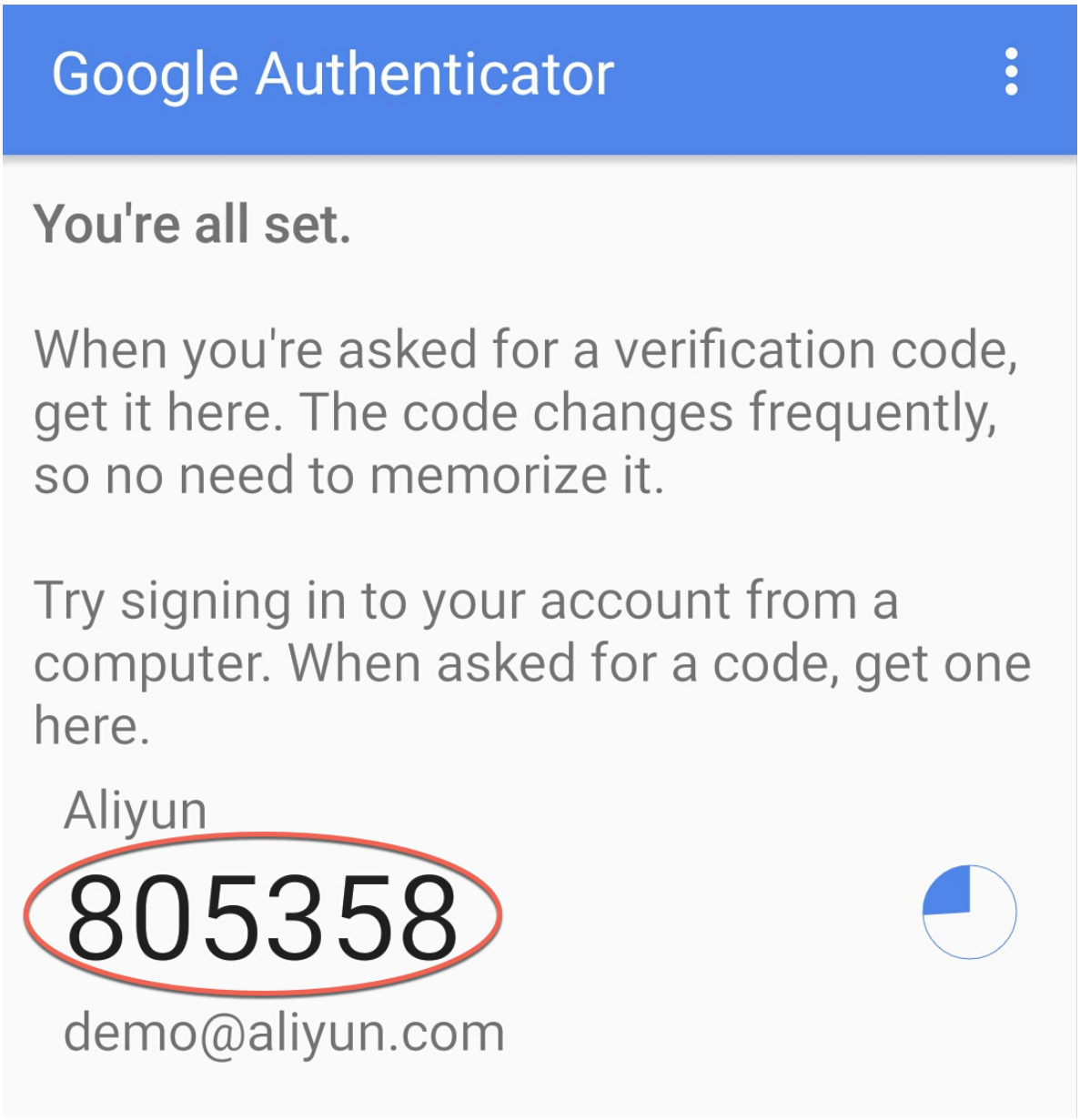
2. Select **Scan a barcode** and then scan the barcode generated on the MFA binding page.

Figure 8-6: Scan a barcode



3. After scanning the code, you will see the window as shown in the following figure that displays your account name and MFA key.

Figure 8-7: Verification



4. On the MFA page, enter the two consecutive MFA codes and then click **Confirm to bind** to bind the authenticator.

Figure 8-8: Confirm to bind